NASA CR-112239-3

USER MANUAL FOR STREAMTUBE CURVATURE ANALYSIS

NASA-CR-112239-3) USER MANUAL FOR STREAMTUBE CURVATURE ANALYSIS: ANALYTICAL METHOD FOR PREDICTING THE PRESSURE DISTRIBUTION ABOUT A NACELLE AT (General Electric Co.) 304 p HC [redacted] CSCL 20D

N73-24315

G3/12 Unclas 03966

ANALYTICAL METHOD FOR PREDICTING THE PRESSURE DISTRIBUTION ABOUT A NACELLE AT TRANSONIC SPEEDS

by J.S. Keith, D.R. Ferguson, P.H. Heck

Prepared by

General Electric Company
Aircraft Engine Group
Cincinnati, Ohio

for

National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia

NAS1-10804

REPRODUCED BY
NATIONAL TECHNICAL
INFORMATION SERVICE
U.S. DEPARTMENT OF COMMERCE
SPRINGFIELD, VA. 22161

This appendix to the User Manual for the Streamtube Curvature Analysis contains the computer program listing. It should be noted that the listing includes explanatory statements and titles so that the program flow is readily discernable. The computer program listing is in CDC Fortran 2.3 source language form, except for three subroutines, GETIX, GETRLX, and SAVIX, which are in Compose 1.1 language.

```
*DECK MAIN
      OVERLAY(STC,0,0)
      PROGRAM STCA(INPUT,OUTPUT,TAPE5,TAPE6=OUTPUT,
     * TAPE1,TAPE2,TAPE4=TAPE2)
      COMMON /BCOMMN/ PROGM,TAPIN,TAPOT,REF(5),PROGSV,FILIN,FILOT
       LOGICAL      TAPIN,TAPOT,                FILIN,FILOT
      EQUIVALENCE (IPROGM,PROGM)
      COMMON /ADAM01/ NAME(6),ADDRES(6),TITLE(6),IDENT(6)
      COMMON /ADAM02/ ENDJOB,DUM1(2),ENDCRD
      LOGICAL         ENDJOB,        ENDCRD
      COMMON /CBITS / BITS,BLANK
       EQUIVALENCE    (IBLANK,BLANK)
      COMMON /CNTRL / K5(8),CARRY,ICHN
       LOGICAL              CARRY
      COMMON /IXORIG/ IIDUM(21),NM,IIIDUM(11)
      COMMON /KEYS  / KEYA(11),KEYB(11),KODA(22)
       DIMENSION      XKEYA(11)
      EQUIVALENCE (XKEYA(1),KEYA(1))
      COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
       LOGICAL        ERR,ERRMAJ,INERR,PRERR
C
      COMMON /CINNER/ INRCTR,RDUM,NINNER(16),CNVF(16)
      COMMON /CMAXIT/ MAXIT,MAJCTR,GREFIN,EDUM
       LOGICAL                     GREFIN
      COMMON /CPRINT/ PPDUM(6),PDUM(20)
      COMMON /CTAPOS/ RESTRT,ENDBDT,STCFIL,K6SV
       LOGICAL        RESTRT,ENDBDT,STCFIL
      COMMON /CTOLRL/ TOLRL,MAXSWP,CLEN,DS2MX,TOLES2,NSWP,
     *           DS1DMP,DS1MXA,DS1MXB,DS1RMS,ES2MX ,DS1RMO
     *,                SG1MIN,TOLINR
      COMMON /TAPES / NTAPO,NTAPN
      DIMENSION AA(8)
C     DS1DMP= DAMPING FACTOR ON DS1, =0 FOR NO DAMPING,  =1 FOR NOMINAL
C     DS1MXA= MAX-DS1
C     DS1MXB= MAX CALCULATED DS1 BEFORE DAMPING
C     DS1RMS= RMS OF THE CALCULATED DS1-S
C     ES2MX = MAX SL POSITION ERROR AS DETERMINED BY THE FLOW BALANCE
C     DS2MX = MAX CALCULATED SL ADJUSTMENT
C     NSWP  = NUMBER OF LRELAX SWEEPS
      COMMON /SELECT/  LENTRY
      DATA KA/1HA/, KBDY/3HBDY/, STC/3HSTC/
      DATA ITRUE/1HT/

      NTAPO = 1
      NTAPN = 2
      WRITE (6,7760)
 7760 FORMAT(1H1,22X,28H* * C A R D    I N P U T * *//)
C     INITIALIZE--- AFTER READING NAMELISTS ID,DIP
      ENDFILE 5
      REWIND 5
 7777 FORMAT(1H1)
 7778 FORMAT(8A10)
 7775 READ (5,7778) AA
      IF( EOF,5 ) 7781,7776
 7776 WRITE (6,7778) AA
      GO TO 7775
 7781 REWIND 5
      READ(5,1001) NAME
```

```
      READ(5,1001) ADDRES
      READ(5,1001) IDENT
 1001 FORMAT (1X,6A10)
      READ (5,1002) IN1,PROGM,TAPIN,TAPOT
 1002 FORMAT (I2,1X,A10,L1,9X,L1)
   11 WRITE (6,1100) PROGM,TAPIN,TAPOT
 1100 FORMAT (1H1,10X,16HEXECUTING PROGM=,A6/10X,6HTAPIN=,L2,5X,
     * 6HTAPOT=,L2/)
      XKEYA(4)= PROGM
      PROGSV= PROGM
      ENDCRD= .FALSE.
      ERRMAJ= .FALSE.
      PRERR = .FALSE.
      DO 2 I=1,3
      KEYA(I)= IBLANK
    2 KEYB(I)= IBLANK
    3 FILIN = TAPIN
      FILOT = TAPOT
      TAPIN = .FALSE.
      TAPOT = .FALSE.
      ERR   = .FALSE.
      DATA IBDY/3HBDY/
      K5    = IBDY
    4 PROGM = BITS
    8 K5    = KA
      GO TO 12
C     CONSECUTIVE DIP LIST READ
    5 READ (5,1003) IN1,IN2,IN3,IN4
 1003 FORMAT(I2,1X,3A10)
      IF(EOF,5) 19,7

    7 GO TO (20,9,10),IN1
    9 K5    = KBDY
      K5(2) = IN3
      ICHN  = IN4
      GO TO 12
   10 K5    = IN2
      K5(2) = IN3

C     INPUT SECTION----- ENTRY STCN TO (1,0)
   12 LENTRY= 1
      LOVER = 1
      CALL OVERLAY(3HSTC,1,0,6HRECALL)
      IF((.NOT.INERR) .AND. (.NOT.ERR) ) GO TO 5
   15 WRITE (6,1004) LOVER,LENTRY
 1004 FORMAT (//2X,9HERR    = T,5X,7HERRCOD=,I2,5X,7HLENTRY=,I2)
      CALL ERRORK(6HERR=T )
      WRITE (6,1000)
 1000 FORMAT(1H1//10X,26H***** JOB TERMINATED *****)
      STOP
   19 ENDJOB= .TRUE.

C     INPUT PROCESSING COMPLETE-- BUILD TABLES
   20 LENTRY= 2
      LOVER = 1
      CALL OVERLAY(3HSTC,1,0,6HRECALL)
      IF(ERR) GO TO 15
      WRITE (6,1140)
```

```
 1140 FORMAT(1H1//17X,3H***,17X,19HSOLUTION    HISTORY,20X,3H***/
     1         7X,103HGRID        +       INNER        +       ORTHOGONALIZAT
     2ION       +         FLOW          +  MATRIX  SOLUTION  /
     3         4X,10HREFINEMENT,9X,10HITERATIONS,41X,7HBALANCE//
     4         2X,106HNREFIN   GRID      INRCTR   CNVF      RMS-DS1    MAX-D
     5S1    MAX-DS1     LIM-ES2     MAX-ES2     MAX-DS2  NSWEEPS  /13X,3HPTS
     6       ,23X,16H(BEFORE DAMPING),4X,7H(AFTER))


C     INITIAL CALCULATION OF DISTANCE ALONG STREAMLINE                GE 205
  205 LENTRY= 1
      LOVER = 3
      CALL OVERLAY(3HSTC,3,0,6HRECALL)
      IF(ERR) GO TO 15
      IF(.NOT.RESTRT) GO TO 210
      RESTRT= .FALSE.
      GREFIN= .TRUE.
      LENTRY= 5
      GO TO 216
C
C     REFINE GRID-- ON FIRST ENTRY INSERT BOWSHOCK                    GE 210
  210 LENTRY= 2
      LOVER = 3
      CALL OVERLAY(3HSTC,3,0,6HRECALL)
      IF(ERR) GO TO 15
      IF( .NOT. GREFIN ) GO TO 2511
      MAJCTR= MAJCTR+1
      INRCTR= 0
      DS2MX = 0.
      NSWP  = 0

C     CALCULATE CURV,PHI,DS1-S.-- ORTHOGONALIZE. ADJUST FLOWS         GE 215
  215 LENTRY= 3
  216 LOVER = 3
      CALL OVERLAY(3HSTC,3,0,6HRECALL)
      IF(ERR) GO TO 15
C     CALCULATE B,RHS-- FLOW BALANCE-- STATION LOOP-- ENTRY-STCB
  225 LENTRY= 1
      LOVER = 2
      CALL OVERLAY(3HSTC,2,0,6HRECALL)
      IF(ERR) GO TO 15
      ES2LIM= SG1MIN*TOLINR
      WRITE (6,1010) MAJCTR,NM,INRCTR,CNVF(MAJCTR),DS1RMS,DS1MXB,DS1MXA,
     1         ES2LIM,ES2MX,DS2MX,NSWP
 1010 FORMAT(I6,I10,I9,F9.2,F12.6,2F10.6,3F11.6,I6)
      IF(INRCTR.EQ.0 .OR. (ES2MX.GE.ES2LIM.AND.INRCTR.LT.NINNER(MAJCTR))
     1   ) GO TO 240
      IF(MAJCTR.LT.MAXIT .AND. GREFIN) GO TO 210
 2511 IF(ES2MX.LT.(CLEN*TOLES2) .OR. INRCTR.GE.NINNER(MAJCTR)) GO TO 300

C     CALCULATE POINT MOVEMENT  (LRELAX)                              GE 240
  240 LOVER = 4
      CALL OVERLAY(3HSTC,4,0)
      IF(ERR) GO TO 15
C     ADJUST STREAMLINES, CALCULATE FAR FIELD VELOCITY DISTRIBUTION   250
  250 LENTRY= 4
      LOVER = 3
      CALL OVERLAY(3HSTC,3,0,6HRECALL)
```

```
      IF(ERR) GO TO 15
      INRCTR= INRCTR+1
      GO TO 215

C     WRITE OUTPUT                                                    GE  300
  300 LENTRY= 2
      LOVER = 2
      CALL OVERLAY(3HSTC,2,0,6HRECALL)
      IF(PDUM(10).EQ.2.) CALL EDUMPS
      IF(ERR) GO TO 15
      IF(ENDJOB) GO TO 100
      IF( IN3.EQ.ITRUE ) TAPIN=.TRUE.
      IF( IN4.EQ.ITRUE ) TAPOT=.TRUE.
       IPROGM= IN2
      GO TO 11
C
  100 WRITE (6,2000)
 2000 FORMAT (1H1//10X,26H********* ENDJOB *********)
      STOP
      END
```

```
*DECK USECDG
       BLOCK DATA USECDG
*USECDG         REPLACE LFIELD USE CARDS
      COMMON /ALLCOM/ C1(24)
      COMMON /CPRINT/ C32(26)
      COMMON /CTHICK/ C7(120)
      COMMON /CIDEX / C5(6)
      COMMON /CFRFIN/ C3(6)
      COMMON /CBEAM2/ C30(20)

      COMMON /CDS2  / C12(900)
      COMMON /CRHS  / RHS(768)

      COMMON /CHDATA/ C9(2200)
      COMMON /CEND  / C2(2)
      COMMON /CCURV / CURV(768)
      COMMON /CPHI1 / PHI1(768)
      COMMON /CS1   / S1(768)
      COMMON /CS2   / S2(768)
      COMMON /SLTAB / C8(384)
      COMMON /CM    / JMS(768)

      COMMON /CB    / B(768)
      COMMON /CZ    / Z(768)
      COMMON /CR    / R(768)
      COMMON /CVM   / VM(768)
      COMMON /CFRFLD/ C4(830)
      COMMON /ERASE2/ C31(1536)
      END
```

```
*DECK STCBLK
      BLOCK DATA STCBLK
*STCBLK         STC  BLOCK DATA                                    -STCBLK-

      COMMON /ALLCOM/ MACHA,PSA,TSA,PTA,TTA, AXIA,RGA,GAMA,
     1                MACHC,PSC,TSC,PTC,TTC, AXIC,RGC,GAMC,
     2                DAXIT,SCALEA,TTE,CHOTST
        REAL          MACHA(1),MACHC
        LOGICAL       AXIA,AXIC,CHOTST
C     TABLE OF INDEX LIMITS
      COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
     *                LO,LESTA,  LDUM(8),
     *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
     *                LEO,LEE, LRO,LRE,LRD
        DIMENSION     LIMITS(24)
        EQUIVALENCE   (LIMITS,LHO)
      COMMON /CBEND / NBCB(2),ANGE(2),CURVE(2),FB(2)
      COMMON /CBITS / BITS,BLANK
      COMMON /CCUBE / NBC(2),C1(2),C2(2),FEND(2)
      COMMON /CGRAV / CG
      COMMON /CPI   / PI,TWOPI,PIQ2,PIQ4,TODEG,TORAD
      COMMON /CPRINT/ PPK(26)
      COMMON /BENDIN/ NBCIN(2),ACF(2)
      COMMON /CBOW  / BSHOCK,DUMBS(8)
        LOGICAL       BSHOCK
      COMMON /CCRX  / CRXSL,CRXOL,CRXSS,CRXE,CRXC,CRMACH
        DIMENSION     CRX(6)
        EQUIVALENCE   (CRX,CRXSL)
C       CRXSL = NEW SL EXTENSION CRITERIA
C       CRXOL = NEW OL EXTENSION CRITERIA
C       CRXSS = EXTENSION CRITERIA FOR NEW OL IN REGION WITH SOME SS-FLOW
C       CRXE  = EXTENSION CRITERIA FOR NEW OL WHICH CROSSES SONIC LINE
C       CRXC  = EXTENSION CRITERIA FOR NEW OL WHICH CROSSES SHOCK WAVE
      COMMON /CFB2  / PASS1
        LOGICAL       PASS1
      COMMON /CINNER/ INRCTR,RDUM,NINNER(16),CNVF(16)
      COMMON /CISBOT/ FARFLD(2),FREE(2),PRES(2),RFF,NZP,
     1                ZP(10),PPS(10), A1,A2,ADUM(6)
        INTEGER       FARFLD,FREE,PRES
      COMMON /CIADIN/ RHOBAS,RHOAMP,IADM
      COMMON /CLBL  / LBL,LSS(2),LBLCTR,MAXLBL,TOLLBL,ES2LBL,SSOL
        LOGICAL       LBL,                                     SSOL
      COMMON /CLINES/ LINES,OMITFK,PTITLE(6)
      COMMON / CNORM / RHL,RM,AHL,ARM
      COMMON /CPRPRN/ PRPRN
        INTEGER       PRPRN
      COMMON /CPTMOV/ VELPOT,ICOB,NODENS,CPTDUM
        LOGICAL       VELPOT
      COMMON /CREFIN/ SLS,SG21,VMG1,VMG2, NGR,NGZ,SGR(10),GR(10),
     1                SGZ(10),GZ(10)
        DIMENSION     G40(40)
        EQUIVALENCE   (G40,SGR)
      COMMON /CSS   / SSFML,SSEF,SSEANG,SSDF,SSFEND,SSFND1
     1               ,SSDLE,A4FACT,BRLX,CURRLX,TSIC,RHOC,RHOCSS
        INTEGER       SSFML
        LOGICAL              SSEF,       SSDF,   SSDLE
C       SSFML = SUPERSONIC CURVATURE FORMULA NUMBER
C       SSEF  = SUPERSONIC ENTERING FLOW, T OR F
```

```
C      SSEANG= ENTERING FLOW ANGLE (DEGREES) FOR SSEF=T
C      SSDF  = SUPERSONIC DISCHARGE FLOW, T OR F
C      SSFEND= SUPERSONIC BEAM DOWNSTREAM END CONDITION, =0,1 FOR PARABOL
C      SSFND1= SUPERSONIC BEAM UPSTREAM END CONDITION, =0,1, FOR PARABOLA
C      SSDLE = SS FLOW BELOW AND AFT OF LE PT, T OR F
C      A4FACT= CENTRAL POINT INFLUENCE COEFFICIENT FACTOR
C      BRLX  = B-RELAXATION FACTOR
C      CURRLX= CURVATURE RELAXATION FACTOR
       COMMON /CSLC  / BRANCH(4)
       COMMON /CTOLRL/ TOLRL,MAXSWP,CLEN,DS2MX,TOLES2,NSWP,
      1                DS1DMP,DS1MXA,DS1MXB,DS1RMS,ES2MX,DS1RMO
      *,               SG1MIN,TOLINR
       COMMON /SLTAB2/ PTR(128)
       COMMON /LINMAX/ LMAX


       DATA MACHA/1.E+15/,PSA,PTA/2*14.696/,TSA,TTA/2*518.7/,
      1     RGA/1716.2/, GAMA/1.4/,            AXIA/.TRUE./
       DATA CHOTST/.TRUE./, SCALEA/1./, TTE/0./
       DATA NBCIN/2,2/

       DATA LHO,LHE/1,0/
       DATA LEO,LEE/1,0/, LRO,LRE/1,0/,MO,NM/1,0/
       DATA MAXNJ/128/, MAXOL/96/, NFCOLS/20/

       DATA BITS/1.E+15/,BLANK/1H /
       DATA PI/3.14159265/, TWOPI/6.2831853/, PIQ2/1.57079632/,
      1     PIQ4/.78539816/, TODEG/57.2957795/, TORAD/.0174532925/

       DATA CG/32.174/
       DATA (PTITLE(I),I=1,6)/6H      ,6H  STRE,6HAMTUBE,6H CURVA,
      * 6HTURE P,6HROGRAM/
       DATA PPK/26*0./

       DATA BSHOCK/F/
       DATA CRX/.375,.375,.125,0.,0.,0./

       DATA NINNER/16*10/,CNVF/16*1./
       DATA (FARFLD(I),I=1,2)/10HFF        ,10H          /
       DATA (FREE(I),I=1,2)/10H          ,10H          /
       DATA (PRES(I),I=1,2)/10H          ,10H          /
       DATA A1,A2/2.,8./,ADUM/.25,5*0./
       DATA RHOBAS,RHOAMP,IADM/.5,.5,0/
       DATA PASS1/.TRUE./
       DATA LBL/.FALSE./, MAXLBL/5/, TOLLBL/.01/
       DATA RN/0./
       DATA RHL/1./, RM/1./
       DATA PRPRN/0/
       DATA VELPOT/F/,ICOB/0/, NODENS/0/
       DATA G40/40*1.E+15/,NGR,NGZ/1,0/
        DATA VMG1,VMG2/.1,.1/, SGR/10.,9*0./, SLS,SG21/.01,1./
       DATA SGZ/10*0./

       DATA SSFML/1/, SSEF/.FALSE./, SSEANG/0./, SSDF/.FALSE./
      1,    SSFEND,SSFND1/.75,.75/
      2,     SSDLE/F/, A4FACT/.3/, BRLX,CURRLX/1.,1./, TSIC/2./
      3,    RHOC,RHOCSS/1.,1./
       DATA BRANCH/4*999./
```

7

```
DATA TOLRL/1.E-3/, MAXSWP/200/ , TOLES2/1.E-3/
DATA SG1MIN/10.E06/, TOLINR/.05/
DATA DS1DMP/.02/, DS1RM0/0./
DATA PTR/128*1./

DATA LMAX/64/
DATA PSA,PTA,TSA,TTA,RGA/5*1./
END
```

```
*DECK EDUMPS
      SUBROUTINE EDUMPS
*EDUMPS         NORMAL TERM. EDUMP
      COMMON /CHDATA/ X1F(1)
       DIMENSION      X1(1)
       EQUIVALENCE    (X1F,X1)
      COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
     *                LO,LESTA,LSO,LSE,LDUM(6),
     *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
     *                LEO,LEE, LRO,LRE,LRD
       DIMENSION      LIMITS(24)
       EQUIVALENCE    (LIMITS,LHO)
      COMMON /CTABPR/  I1TAB
      I1TAB = LFO
      CALL TABPRT(6HCADJWF,X1F,LFE,8)
      I1TAB = LO
      CALL TABPRT(6HSTATAB,X1,LESTA,5)
      RETURN
      END
```

```
*DECK ERRORK
      SUBROUTINE ERRORK(NAME)
      COMMON /ALLCOM/ MACHA,PSA,TSA,PTA,TTA, AXIA,RGA,GAMA,
     1                MACHC,PSC,TSC,PTC,TTC, AXIC,RGC,GAMC,
     2                DAXIT,SCALEA,TTE,CHOTST
        REAL          MACHA(1),MACHC
        LOGICAL       AXIA,AXIC
        LOGICAL       CHOTST
      COMMON /ERASE2/ AREA(96),AREAO(96),DISP(96),PT(96),LAMBDA(96),
     1                RHO(96),SQRTVV(96),TS(96),TT(96),VMSQ(96),
     2                VVKQKP(96),
     2                WQA(96),WSTA(96), RG(96),C2CP(96),FGR(96)
        REAL          LAMBDA
        DIMENSION     ES2(96),SDNQRM(96)
        EQUIVALENCE   (ES2,VVKQKP),(SDNQRM,RHO)
        DIMENSION     RCU(96)
        EQUIVALENCE   (RCU,LAMBDA)
C     FIELD TABLES
C      INDEX- M=MO,NM
      COMMON /CZ    / Z(300)
      COMMON /CR    / R(300)
      COMMON /CS2   / S2(300)
      COMMON /CS1   / S1(300)
      COMMON /CPHI1 / PHI1(300)
      COMMON /CM    / JMS(300)
      COMMON /CCURV / CURV(300)

      COMMON /CB    / B(300)
      COMMON /CIDEX / M,J,MU,MD,ISTAG
C     TABLE OF INDEX LIMITS
      COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
     *                LO,LESTA,  LDUM(8),
     *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
     *                LEO,LEE, LRO,LRE,LRD
        DIMENSION     LIMITS(24)
        EQUIVALENCE   (LIMITS,LHO)
      COMMON /CVM   / VM(300)

C     STREAMLINE TABLE
      COMMON /SLTAB / W(128),X2(128),SLCHN(128)
        INTEGER SLCHN
C     BOUNDARY TABLE
C      INDEX- LB=LBDO,LBDE
C      LBNEXT= INCREMENT TO NEXT BOUNDARY
C      LBZ1  = INCREMENT TO THE FIRST BOUNDARY POINT (=0 BEFORE COALLATIO
C      CHNAME= CHANNEL WITH WHICH THE BOUNDARY DATA IS ASSOCIATED
C      UP    = T OR F FOR UPPER OR LOWER BOUNDARY
C      LEDEX = RELATIVE INDEX OF L.E. POINT WHEN LOWER AND UPPER SURFACE
C              CONTOURS ARE CONNECTED
C      BDNAME,LBA,LBB=NAME AND INDEX LIMITS OF SPECIFIC BOUNDARY
C                     DATA WHEN BOUNDARIES ARE COALLATED
      DIMENSION       BDT(1),LBNEXT(1),LBZ1(1),
     1                CHNAME(1),UP(1),LEDEX(1),
     2                ZBT(1),RBT(1),ANGBT(42)
        LOGICAL       UP
        INTEGER BDT,CHNAME,BDNAME
        DIMENSION     BDNAME(1),LBA(1),LBB(1)
        EQUIVALENCE   (BDNAME,ZBT), (LBA,RBT), (LBB,ANGBT)
```

```
C     FLOW ADJUSTMENT TABLE
C      INDEX- LF=LFO,LFE
C      NFCOLS= 8
C      X1F    = ORTHOGONAL COORDINATE
C      X2F    = STREAMLINE COORDINATE OF SL EMINATING FROM T.E.
C      X1BF   = X1-COORDINATE OF CHOKE STATION OF FLOW BELOW T.E.
C      X1AF   = X1-COORDINATE OF CHOKE STATION OF FLOW ABOVE T.E.
C      S1F    = S1-COORDINATE OF T.E. (UPPER SURFACE). THIS ITEM
C               IS USED WHEN INTERPOLATING FOR WAKE DELTA-STAR.
C      LFB,LFA=INDICES OF STATIONS BELOW AND ABOVE T.E.
C      NCHB,NCHA=NUMBER OF CHANNELS BELOW AND ABOVE T.E.
C      LRF    = INDEX OF DUMMY ORTCHN LIST FOR THE T.E.
C      LRXF   = INDEX OF LAST CHANNEL BELOW THE T.E.
C      JORDER= 0 IF TOTAL FLOW AT X1F IS GIVEN
C            = 2 IF FLOW ABOVE T.E. IS GIVEN
C            = 1 IF FLOW BELOW T.E. IS GIVEN
C      JORDER= -1 IF FLOW AT X1F IS CHOKED AND SINGLE CHANNEL
       DIMENSION       X1F(1),X2F(1),X1BF(1),X1AF(1),
      1                S1F(1),NCHB(1),NCHA(1),JORDER(1),VNR(12)
        EQUIVALENCE    (LFB,X1BF),(LFA,X1AF),(LRF,NCHB),(LRXF,NCHA)
        DIMENSION      LFB(1),LFA(1),LRF(1),LRXF(1)
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
       COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
      1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
      1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
      3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
        LOGICAL        PRIM
        DIMENSION      SCHOKE(1)
        EQUIVALENCE    (SCHOKE,DWDV)

        EQUIVALENCE   (BDT,X1F,X1), (LBNEXT,X2F,LNEXT), (LBZ1,X1BF,MLB)
        EQUIVALENCE   (CHNAME,X1AF,MUB), (UP,S1F,PRIM)
        EQUIVALENCE   (LEDEX,NCHB,TYPELB), (ZBT,NCHA,NAMELB)
        EQUIVALENCE   (RBT,JORDER,ILB), (ANGBT,VNR,FLB)

       COMMON /CTABPR/ I1TAB

       WRITE (6,100) NAME
  100 FORMAT(//2X,13HERRORK CALL--,1A6//)

       CALL TABPRT(6HALLCOM,MACHA,20,8)
       I1TAB = LBDO
       CALL TABPRT(6HBDYTAB,BDT,LBDE,3)
       I1TAB = LFO
       CALL TABPRT(6HCADJWF,X1F,LFE,8)
       I1TAB = LO
       CALL TABPRT(6HSTATAB,X1,LESTA,5)
  150 WRITE (6,1150) (J,X2(J),SLCHN(J),W(J),J=1,NJ)

       CALL JMSPRT
       CALL TABPRT(2HS1,S1,NM,10)
       CALL TABPRT(2HS2,S2,NM,10)
       CALL TABPRT(1HZ,Z,NM,10)
       CALL TABPRT(1HR,R,NM,10)
```

11

```
      CALL TABPRT(4HPHI1,PHI1,NM,10)
      CALL TABPRT(4HCURV,CURV,NM,10)
      CALL TABPRT(2HVM,VM,NM,10)
      CALL TABPRT(1HB,B,NM,10)
      CALL TABPRT(6HERASE2,AREA,1536,8)
  160 WRITE (6,1160) NQ
 1160 FORMAT(///4H NQ=,I4)
      CALL TABPRT(2HZQ,ZQ,NQ,10)
      CALL TABPRT(2HRQ,RQ,NQ,10)
      CALL TABPRT(5HCURVQ,CURVQ,NQ,10)
      CALL TABPRT(5HPHI1Q,PHI1Q,NQ,10)
      CALL TABPRT(3HX2Q,X2Q,NQ,10)
      CALL TABPRT(5HKCHNQ,KCHNQ,NQ,10)

      LSTOP = 5
      IF(LSTOP.EQ.5) STOP
      RETURN
 1150 FORMAT(///1X17HSTREAMLINE TABLE-/17X32HJ        X2              SLCHN
     *     W/(I18,F12.6,6X,A6,F12.6,),)
      END
```

```
*DECK ATAN3
      FUNCTION ATAN3(DY,DX,ANGREF)
*ATAN3-         ARCTAN FUNCTION WITH REFERENCE ANGLE          -ATAN3-

C     LIMITS ARE-          (-PI) .LE. (ATAN3-ANGREF) .LT. (+PI)

      COMMON /CATAN3/ DANG
      COMMON /CPI    / PI,TWOPI
      DATA KNAME/6HATAN3 /

      ANG    = ATAN2(DY,DX)
      N      = 20
   50 N      = N-1
      IF(N.EQ.0) CALL ERRORK(KNAME)
      DANG   = ANG-ANGREF
      IF(PI-DANG) 60,70,70
   60 ANG    = ANG-TWOPI
      GO TO 50
   70 IF(DANG+PI) 80,90,90
   80 ANG    = ANG+TWOPI
      GO TO 50
   90 ATAN3 = ANG
      RETURN
      END
```

```
*DECK BARC
      SUBROUTINE BARC(I)
*BARC--          BOUNDARY INTERVAL CURVALINEAR DIST          -BARC-

C     INPUT-
C      BDY    = BOUNDARY TABLE OF Z,R,ANG
C      I      = INDEX OF COOR-Z RELATIVE TO BDY-TABLE ORIGIN

C     OUTPUT-
C      DR     = DELTA-R = R(IV+1)-R(IV)
C      DZ     = DELTA-Z = Z(IV+1)-Z(IV)
C      DX     = CHORD CONNECTING THE POINTS OF THE INTERVAL
C      ANGCHD= ANGLE OF THE CHORD
C      YPA    = ANGLE RELATIVE TO THE CHORD, POINT-IV
C      YPB    = ANGLE RELATIVE TO THE CHORD, POINT-IV+1
C      SINTVL= CURVALINEAR DISTANCE BETWEEN POINTS IV,IV+1
C      (ALSO-YPASQ,YPBSQ,YPAB)

C     BOUNDARY TABLE
C      INDEX- LB=LBDO,LBDE
C      LBNEXT= INCREMENT TO NEXT BOUNDARY
C      LBZ1   = INCREMENT TO THE FIRST BOUNDARY POINT (=0 BEFORE COALLATIO
C      CHNAME= CHANNEL WITH WHICH THE BOUNDARY DATA IS ASSOCIATED
C      UP     = T OR F FOR UPPER OR LOWER BOUNDARY
C      LEDEX = RELATIVE INDEX OF L.E. POINT WHEN LOWER AND UPPER SURFACE
C              CONTOURS ARE CONNECTED
C      BDNAME,LBA,LBB=NAME AND INDEX LIMITS OF SPECIFIC BOUNDARY
C                     DATA WHEN BOUNDARIES ARE COALLATED
       COMMON /CHDATA/ BDT(1),LBNEXT(1),LBZ1(1),
      1                CHNAME(1),UP(1),LEDEX(1),
      2                ZBT(1),RBT(1),ANGBT(42)
         LOGICAL       UP
         INTEGER BDT,CHNAME,BDNAME
         DIMENSION     BDNAME(1),LBA(1),LBB(1)
         EQUIVALENCE   (BDNAME,ZBT), (LBA,RBT), (LBB,ANGBT)
       COMMON /CBEAM2/ DR,DZ,YPA,YPB,F,G, DX,YQDX,ZM,RM,ANGM,CURVM,S1M,
      1                RZONLY, ANGCHD,SINTVL, YPASQ,YPAB,YPBSQ
         LOGICAL       RZONLY

       DZ     = ZBT(I+3)-ZBT(I)
       DR     = RBT(I+3)-RBT(I)
       DX     = SQRT(DZ*DZ+DR*DR)
       IF(DX.EQ.0.) GO TO 90
       ANGCHD= ATAN3(DR,DZ,ANGBT(I))
       YPA    = ANGBT(I)-ANGCHD
       YPB    = ANGBT(I+3)-ANGCHD
       YPASQ  = YPA*YPA
       YPAB   = YPA*YPB
       YPBSQ  = YPB*YPB
    90 SINTVL= DX*(1. + (YPASQ-.5*YPAB+YPBSQ)/15.)

       RETURN
       END
```

```
*DECK BEAM
      SUBROUTINE BEAM(X,Y,ANG,N)
*BEAM--          ROTATED CUBICS SIMILATING A BEAM                -BEAM-
C                FIT TO COORDINATE POINTS

      DIMENSION X(100),Y(100),ANG(100)

C     INPUT-
C      X,Y    = COORDINATES OF POINTS
C      ANG    = ESTIMATED ANGLE AT THE GIVEN POINTS, RADIANS  (MA=1)
C      ANG(1)= ESTIMATED ANGLE AT THE FIRST POINT (MA=0)
C      N      = NUMBER OF POINTS
C      MA     = 0 IF THE VALUES OF ANGLES ARE NOT ESTIMATED,
C             = 1 IF ESTIMATED ANGLES ARE GIVEN
C      MB     = NO OF ITERATIONS
C      KD     = STORAGE INCREMENT OF  X,Y,ANG
C      KORDER= 0 IF ERROR1 IS TO BE CALLED WHEN PTS ARE OUT OF ORDER
C             = -1 TO SKIP THE POINT ORDER CHECK
C             = .GE.1    IF RETURN IS TO BE MADE FOR CORRECTIVE ACTION
C      (IF NOT INPUT MA=0, MB=1, KD=1, AND KORDER=0)
C      SUBROUTINE BEND MUST BE PROVIDED TO CALCULATE THE FOLLOWING COEFFI
C              A(2,1),A(3,1),B(1), A(1,N),A(2,N),B(N)

C     OUTPUT-
C      ANG    = CALCULATED VALUE OF THE CURVE ANGLE, RADIANS
C      B      = SLOPE IN ROTATED COORDINATES, LEFT END OF SEGMENT
C      YPB    = SLOPE IN ROTATED COORDINATES, RIGHT END OF SEGMENT
C      ACHD   = ANGLE (RELATIVE TO HORIZONTAL) OF THE LINE SEGMENTS, RADIA
C      CHD    = LENGTHS OF THE LINE SEGMENTS BETWEEN THE INPUT POINTS, CHO
C      KORDER= INDEX OF 2ND OF ADJACENT OUT-OF-ORDER PTS, NOT=0 ON ENTRY

C     NOTE-COMMON /ERASE/ MUST BE 8*N IN LENGTH.  ITS LENGTH MAY BE CHANG
C      BY A $USE CARD WITHOUT PROGRM RECOMPILATION.

C     ORDER OF STORAGE IN COMMON /ERASE/ IS - A(1,3),A(1,1),A(1,2),B(1),
C       YPB(1),DA(1),ACHD(1),CHD(1), A(2,1),A(2,2),A(2,3),B(2),YPB(2),DA(

      COMMON /CATAN3/ DANG
      COMMON /CREAM / MA,MB,KD
      COMMON /ERASE / A(3),B(1),YPB(1),DA(1),ACHD(1),CHD(793)
       DIMENSION      YPA(100)
       EQUIVALENCE    (YPA,B)

      IF(N.LE.1) CALL ERROR1
      M      = MA
      N8     = 8*N-7

C     CALCULATE THE CHORDS CONNECTING THE GIVEN POINTS
C       AND CALC THE TURNING ANGLES BETWEEN SUCCESSIVE CHORDS
      K      = 1
      I      = 1
      IM8    = 1
      ACHD(1)=ANG(1)
  100 KP     = K+KD
      SX     = X(KP)-X(K)
      SY     = Y(KP)-Y(K)
      B(I)   = ANG(K)
      CHD(I)= SQRT(SX*SX+SY*SY)
```

15

```
      ACHD(I)=ATAN3(SY,SX,ACHD(IM8))
      DA(I) = DANG
      IF(I.GT.9 .AND. (ABS(DA(I))+ABS(DA(IM8))).GT.PI .AND.
     * KORDER.NE.(-1)) GO TO 800
  130 IM8   = I
      I     = I+8
      K     = K+KD
      IF(I-N8) 100,140,140
  140 ACHD(I)=ACHD(I-8)
      DA(I) = 0.
      B(I)  = ANG(K)

C     SLOPES IN THE ROTATED COORDINATE SYSTEM
C      FROM THE ESTIMATED INPUT ANGLES
      I     = 1
      IF(M) 160,180,160
  160 YPA(I)= TAN(B(I)-ACHD(I))
      YPB(I)= TAN(B(I+8)-ACHD(I))
      I     = I+8
      IF(I-N8) 160,200,200

C     SLOPES EQUAL TO A FRACTION OF THE LINE SEGMENT TURNING
  180 YPA(1)= -.2*DA(9)
      I     = 9
  185 YPB(I-8)=.4*DA(I)
      YPA(I)= -YPB(I-8)
      I     = I+8
      IF(I-N8) 185,190,190
  190 YPB(I-8)=.2*DA(I-8)

C     END EQUATIONS
  200 CALL BEND(N)

C     MATCHING  ANGLE  AND  CURVATURE  EQUATIONS
      IF(N-2) 250,300,250
  250 I     = 9
      GO TO 260
  255 A(I)  = CHD(I)*(1.+1.5*YPA(I)*YPA(I))
      A(I+2)= CHD(I-8)*(1.+1.5*YPB(I-8)*YPB(I-8))
      A(I+1)= 2.*(A(I)+A(I+2))
      B(I)  = -2.*A(I  )*DA(I) - A(I+2)*DA(I+8)
      I     = I+8
  260 IF(I-N8) 255,300,300

C     ROUTINE TDSEQ  -  TRIDIAGONAL SIMULTANEOUS EQUATIONS
C       SOLUTION TO  AX=B.  ON RETURN SOLUTION VECTOR  X  IS STORED IN  B
  300 A(3)  = A(3)/A(2)
      B(1)  = B(1)/A(2)
      I     = 9
C          SPECIAL LOGIC FOR A(1,3)
           A(1)  = A(1)/A(2)
           A(10) = A(10)-A(9)*A(3)
           A(11) = (A(11)-A(9)*A(1))/A(10)
           GO TO 312
  310 A(I+1)= A(I+1)-A(I)*A(I-6)
      A(I+2)= A(I+2)/A(I+1)
  312 B(I)  = (B(I)-A(I)*B(I-8)) / A(I+1)
      I     = I+8
```

```
      IF(I-N8) 310,320,340
C          SPECIAL LOGIC FOR  A(N,N-2)
  320      A(I)   = A(I)-A(I+2)*A(I-14)
           B(I)   = B(I)-A(I+2)*B(I-16)
           GO TO 310
C     BACK SUBSTITUTION
  340 I      = N8
  350 I      = I-8
      IF(I-1) 400,355,360
C          SPECIAL LOGIC FOR A(1,I)
  355      B(1)   = B(1)-A(1)*B(17)
  360 B(I)   = B(I)-A(I+2)*B(I+8)
      GO TO 350

C    REEVALUATE  YPB
  400 I      = 9
  405 YPB(I-8)= B(I)+DA(I)
      I      = I+8
      IF(I-N8) 405,405,450

C    RETURN FOR ANOTHER ITERATION
  450 M      = M+1
      IF(M-MB) 200,200,500

C    ANGLES
  500 I      = 1
      K      = 1
  505 ANG(K)= ACHD(I)+ATAN(B(I))
      I      = I+8
      K      = K+KD
      IF(I-N8)505,505,530
  530 KORDER= 0
      GO TO 900

C     ERROR - OUT OF ORDER POINTS
  800 IF(KORDER.EQ.0) CALL ERROR1
      KORDER= K

  900 RETURN
      END
```

```
*DECK CBEAM
      BLOCK DATA BEAMBK
*CBEAM-          DATA FOR /CBEAM /                              -CBEAM-
      COMMON /CBEAM / MA,MB,KD,KORDER
      DATA MA,MB,KD,KORDER/0,1,1,0/
      END
```

```
*DECK BEND
      SUBROUTINE BEND(NN)
*BEND--          END CONDITIONS FOR THE BEAM FIT                -BEND-

C     ON ENTRY -
C       N       = NUMBER OF POINTS
C        ALSO DEFINED ON ENTRY - IN COMMON/CBEND/ -
C       NBC(L)= BOUNDARY CONDITION INDICATOR FOR LEFT(L=1) AND RIGHT(L=2)
C             = 0, 1, OR 2
C       ANGE(L)=ANGLE IN DEGREES  IF  NBC(L)=1
C       CURVE(L)=CURVATURE  IF NBC(L)=2
C       FEND(L)= RATIO OF SHEAR OF THE END TO NEXT TO END INTERVAL, NBC(L)

C     ON RETURN-
C       COEFFICIENTS - A(2),A(3),B(1)  AND  A(N8),A(N8+1),B(N8)

        COMMON /CBEND / NBC(2),ANGE(2),CURVE(2),FEND(2)
        COMMON /CPI   / PI,TWOPI,PIQ2,PIQ4,TODEG,TORAD
        COMMON /ERASE / A(3),B(1),YPB(1),DA(1),ACHD(1),CHD(793)

C     INITIALIZE
        N       = NN
C       N8      = INDEX FOR RIGHT END POINT
        N8      = 8*N-7
        A(1)    = 0.
        A(2)    = 1.
        A(3)    = 0.
        A(N8)   = 0.
        A(N8+1)=1.
        A(N8+2)=0.

C     A STRAIGHT LINE IS USED FOR N=2 IF  NBC(1)=NBC(2)=0
        NBCS = NBC(1)+NBC(2)
        IF(N.GT.2 .OR. NBCS.GT.0) GO TO 80
        B(1)    = 0.
        B(2)    = 0.
        GO TO 900

C     CHECK IF PARABOLA (F=0) SHOULD BE USED
   80 IF(N.EQ.3 .AND. NBCS.EQ.0) GO TO 90
        F1      = FEND(1)
        F2      = FEND(2)
        GO TO 110
   90 F1      = 0.
        F2      = 0.

C     NBC=01,  Y  AND  ANGLE  SPECIFIED
C       LEFT END
  110 IF(NBC(1).NE.01) GO TO 120
        B(1)    = TAN(TORAD*ANGE(1)-ACHD(1))
C       RIGHT END
  120 IF(NBC(2).NE.01) GO TO 210
        B(N8) = TAN(TORAD*ANGE(2)-ACHD(N8))

C     NBC=02,  Y  AND  CURVATURE  SPECIFIED
C       LEFT END
  210 IF(NBC(1).NE.02) GO TO 220
        A(2)    = 4.
```

```
      A(3)   = 2.
      B(1)   = -2.*DA(9)+CHD(1)*CURVE(1)*(1.+1.5*B(1)*B(1))
C      RIGHT END
  220 IF(NBC(2).NE.02) GO TO 310
      A(N8) = 2.
      A(N8+1)=4.
      B(N8) = -CHD(N8-8)*CURVE(2)*(1.+1.5*YPB(N8-8)*YPB(N8-8))

C     NBC=0,   YPPP = F * YPPP(OF ADJACENT INTERVAL)
C      LEFT END
  310 IF(NBC(1).NE.0) GO TO 320
      IF(N.EQ.2) GO TO 315
      DX1SQ = CHD(1)*CHD(1)
      DX2SQ = CHD(9)*CHD(9)
      A(2)   = DX2SQ
      A(1)   = -F1*DX1SQ
      A(3)   = A(2)+A(1)
      B(1)   = F1*DA(17)*DX1SQ - DA(9)*DX2SQ
      GO TO 320
  315 A(3)   = 1.
      B(1)   = 0.
C      RIGHT END
  320 IF(NBC(2).NE.0) GO TO 900
      IF(N.EQ.2) GO TO 325
      DXNSQ = CHD(N8-8)*CHD(N8-8)
      DXMSQ = CHD(N8-16)*CHD(N8-16)
      A(N8+2)=-F2*DXNSQ
      A(N8+1)=DXMSQ
      A(N8) = A(N8+1)+A(N8+2)
      B(N8) = F2*DA(N8-8)*DXNSQ
      GO TO 900
  325 A(N8) = 1.
      B(N8) = 0.

  900 RETURN
      END
```

```
*DECK CBEND
      BLOCK DATA BENDBK
*CBEND-         DATA FOR /CBEND /                                    -CBEND-
      COMMON /CBEND / NBC(2),ANGE(2),CURVE(2),FEND(2)
      DATA NBC,ANGE,CURVE,FEND/2*0,6*0./
      END
```

```
*DECK BFI
      SUBROUTINE BFI
*BFI---          BEAM FIT INTERPOLATION                              -BFI-

C     INPUT-
C      DR     = R(I+1)-R(I)
C      DZ     = Z(I+1)-Z(I)
C      YPA    = ANGLE RELATIVE TO THE CHORD, POINT-I
C      YPB    = ANGLE RELATIVE TO THE CHORD, POINT-I+1
C      F      = X/DX
C      G      = (DX-X)/DX
C      RZONLY= T  IF  YQDX, RM AND ZM  ONLY ARE TO BE COMPUTED

C     OUTPUT DATA AT THE INTERMEDIATE POINT WITHIN THE INTERVAL
C      YQDX   = Y/DX, DISTANCE NORMAL TO THE CHORD
C      ZM     = Z-Z(I)
C      RM     = R-R(I)
C      DX     = LENGTH OF THE CHORD
C      ANGM   = ANG-ANGCHD
C      CURVM  = CURVATURE
C      S1M    = CURVALINEAR DISTANCE FROM POINT-1

C     NOTES-
C      CHORD = LINE BETWEEN POINTS  I  AND  I+1

       COMMON /CBEAM2/ DR,DZ,YPA,YPB,F,G, DX,YQDX,ZM,RM,ANGM,CURVM,S1M,
      1                RZONLY
        LOGICAL        RZONLY

       DOUBLE PRECISION C1,C2,C3,C4,C5

       YQDX   = F*G*(G*YPA-F*YPB)
       RM     = YQDX*DZ+F*DR
       ZM     = F*DZ-YQDX*DR
       IF(RZONLY) GO TO 990
       DX     = SQRT(DR*DR+DZ*DZ)
       ANGM   = YPA*(3.*G-2.)*G + YPB*(3.*F-2.)*F
       CURVM  = (YPA*(6.*G-2.)+YPB*(-6.*F+2.))/(DX*(1.+1.5*ANGM*ANGM))
       YPASQ  = YPA*YPA
       YPAB   = YPA*YPB
       YPBSQ  = YPB*YPB
       C1     = 1.+.5*YPASQ
       C2     = -2.*YPASQ-YPAB
       C3     = (11.*(YPASQ+YPAB) + YPBSQ+YPBSQ)/3.
       C4     = -3.*YPASQ - 4.5*YPAB - 1.5*YPBSQ
       C5     = 9.*(YPASQ+YPAB+YPAB+YPBSQ)/10.
       S1M    = DX*(F*(C1+F*(C2+F*(C3+F*(C4+F*C5)))))
  990  RETURN
       END
```

```
*DECK CBFI
      BLOCK DATA BFIBLK
*CBFI--          BLOCK DATA FOR BFI                                    -CBFI-
      COMMON /CBEAM2/ DR,DZ,YPA,YPB,F,G, DX,YQDX,ZM,RM,ANGM,CURVM,S1M,
     1                RZONLY
       LOGICAL        RZONLY
      DATA RZONLY/.FALSE./
      END
```

```
*DECK FHEAD
      SUBROUTINE FHEAD(LA1)
CFHEAD---- CDC VERSION
      COMMON /ADAM01/ NAME(6),ADDRES(6),TITLE(6),IDENT(6)
      COMMON /CLINES/ LINES,OMITFK,PTITLE(6)
      COMMON /LINMAX/ LMAX

      LA     = LA1

C     ADJUST LINE COUNT
    5 LINTOT= LINES+LA
      IF( LINTOT.GT.LMAX ) GO TO 8
      LINES = LINTOT
    6 RETURN
C     RESTORE AND PRINT IDENTIFICATION IF LINE COUNT.GT.LMAX
    8 WRITE (6,810) TITLE,PTITLE,IDENT
      LINES = LA+3
      GO TO 6
  810 FORMAT(1H1,6A10,33X,6A6/1X,6A10)
      END
```

```
*          DECK GETIX
           IDENT  GETIX
           ENTRY  GETIX,SAVIX
*       SUBROUTINE GETIX
*
*       COMMON /CM    / JMS(300)
*       COMMON /CIDEX / M,J,MU,MD,ISTAG
*       INPUT-
*        JMS   = ARRAY CONTAINING PACKED INDICES J,MU,MD,ISTAG
*        M     = INDEX OF  -JMS- ARRAY
*       OUTPUT-
*        J     = STREAMLINE NUMBER
*        MU    = M- UPSTREAM
*        MD    = M- DOWNSTREAM
*        ISTAG = INDICATOR FOR STAGNATION POINT, ETC.
*
 GETIX    BSSZ   1
          SA1    M            CONTENTS OF M IN X1
          SA2    X1+JMS-1     JMS(M) IN X2
          SB3    0            INITIALIZE
          SB4    3
 LOOPG    SA3    MASK1+B3     LOAD MASK
          BX6    X2*X3        AND TO MASK
          SA1    SHIFT+B3     SHIFT BITS IN X1
          SB5    X1           MOVE TO B5
          AX6    X6,B5        SHIFT
          SA6    J+B3         STORE
          SB3    B3+1
          LE     B3,B4,LOOPG
          JP     GETIX        TRA FOR RETURN
*
*       SUBROUTINE SAVIX
*       INPUT-
*        M     = INDEX OF JMS ARRAY
*        J     = STREAMLINE NUMBER
*        MU    = M- UPSTREAM
*        MD    = M- DOWNSTREAM
*        ISTAG = INDICATOR FOR STAGNATION POINT, ETC.
*       OUTPUT-
*        JMS(M)= PACKED J,MU,MD,ISTAG
*
 SAVIX    BSSZ   1
          MX3    0
          SB3    0            INITIALIZE
          SB4    3
 LOOPS    SA2    B3+J         J IN X2
          SA1    SHIFT+B3
          SB5    X1
          LX2    X2,B5        SHIFT LEFT
          BX3    X3+X2        OR TO X3
          SB3    B3+1
          LE     B3,B4,LOOPS
          SA1    M
          BX6    X3           MOVE TO X6
          SA6    X1+JMS-1     STORE JMS(M)
          JP     SAVIX        TRA FOR RETURN
 MASK1    DATA   0000000007760000000000
          DATA   0000000000017777000000
          DATA   0000000000000077774
```

25

```
          DATA    00000000000000000003
SHIFT     DATA    28
          DATA    15
          DATA    2
          DATA    0
          USE     /CM/
JMS       BSS     300
          USE     /CIDEX/
M         BSS     1
J         BSS     1
MU        BSS     1
MD        BSS     1
ISTAG     BSS     1
*
          END
```

```
*DECK GETRLX
          IDENT   GETRLX
          ENTRY   GETRLX
*
 GETRLX   BSSZ    1
          SB4     25              INITIALIZE REGISTERS
          SB7     -5
 LOOP     SB7     B7+5            INDEX B7
          GE      B7,B4,GETRLX
 LOOP2    SA1     B7+M            CONTENTS OF M IN X1
          SA2     X1+JMS-1        JMS(M) IN X2
          SA3     MASK1           MU-MASK IN X3
          BX6     X2*X3           EXTRACT MU
          SA1     SHIFT
          SB3     X1              SHIFT BITS
          AX6     X6,B3           SHIFT RIGHT
          NZ      X6,UPO          TEST FOR STREAMLINE ORIGIN
          SA4     M               M- TO X4
          BX6     X4              MOVE TO X6
 UPO      SA6     B7+MU           STORE CURRENT MU
          SA3     MASK1+1         MD-MASK IN X3
          BX6     X2*X3           EXTRACT MD
          SA1     SHIFT+1
          SB3     X1              SHIFT BITS
          AX6     X6,B3           SHIFT RIGHT
          NZ      X6,DNO          TEST FOR STREAMLINE TERMINATION
          SA4     M               M- TO X4
          BX6     X4              MOVE TO X6
 DNO      SA6     B7+MD           STORE CURRENT MD
          SA3     MASK1+2         ISTAG-MASK IN X3
          BX6     X2*X3           EXTRACT ISTAG
          SB6     3
          SB3     X6              MOVE LOW ORDER BITS TO B3
          NE      B3,B6,NOTPO     TEST FOR PARTIAL ORTHOGONAL
          ZR      B7,NOTPO        BRANCH IF MID-POINT
          SB3     5
          EQ      B3,B7,UPPO
          SB3     15
          EQ      B3,B7,UPPO
          SA4     B7+MD           CURRENT MD IN X4
          BX6     X4              MOVE TO X6
          SA6     B7+M            RESET M TO MOVE RIGHT
          JP      LOOP2
*
 UPPO     SA4     B7+MU           CURRENT MU IN X4
          BX6     X4              MOVE TO X6
          SA6     B7+M            RESET M TO MOVE LEFT
          JP      LOOP2
*
 NOTPO    SB3     15
          GE      B7,B3,LOOP      CONTINUE IF ON EXTREME(M2,M6) POINTS
          NZ      B7,TEST1        CONTINUE CHECK
          SA4     MU              MU IN X4
          BX6     X4              MOVE TO X6
          SA6     M+5             SET UP FOR M3,M5
          SA4     MD
          BX6     X4              MOVE TO X6
          SA6     M+10
```

```
          JP      LOOP
*
TEST1     SB3     5
          NE      B7,B3,TEST2
          SA4     B7+MU         SET UP M2 POINT
          BX6     X4            MOVE TO X6
          SA6     M+15
          JP      LOOP
TEST2     SA4     B7+MD         SET UP M6 POINT
          BX6     X4            MOVE TO X6
          SA6     M+20
          JP      LOOP
MASK1     DATA    0000000000001777700000
          DATA    0000000000000000077774
          DATA    0000000000000000000003
SHIFT     DATA    15
          DATA    2
          USE     /CM/
JMS       BSS     300
          USE     /CIDEXR/
M         BSS     2
MU        BSS     1
MD        BSS     1
ISTAG     BSS     21
          END
```

```
*DECK JMSPRT
      SUBROUTINE JMSPRT
*JMSPRT         PRINT INDEX ARRAY,  JMS                          -JMSPRT-

      COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
     *                LO,LESTA,  LDUM(8),
     *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
     *                LEO,LEE, LRO,LRE,LRD
        DIMENSION     LIMITS(24)
        EQUIVALENCE   (LIMITS,LHO)

      COMMON /CIDEX / M,J,MU,MD,ISTAG
      COMMON /CM    / JMS(300)
      COMMON /ERASE / IOUT(800)

C     RESTOR PAGE
      WRITE (6,1000)

      M     = 1
      IS     = 30
   40 I     = 1
      MA    = M
   50 CALL GETIX
      IOUT(I)=J
      IOUT(I+1)=MU
      IOUT(I+2)=MD
      IF(ISTAG .EQ. 0) GO TO 60
      IOUT(IS+1) = M
      IOUT(IS+2) = ISTAG
      IS = IS + 2
   60 I     = I+3
      M     = M+1
      IF(I.LT.30 .AND. M.LE.NM) GO TO 50
      IB    = I-1
      WRITE (6,1002) MA,(IOUT(L),L=1,IB)
      IF(M.LE.NM) GO TO 40
      WRITE (6,1004) (IOUT(I),I=31,IS)
 1000 FORMAT(8H1J-MU-MD)
 1002 FORMAT(1X,I5,30I4)
 1004 FORMAT(/8H M-ISTAG/(6X,20I5))
      RETURN
      END
```

```
*DECK LBF
      FUNCTION LBF(BDYNAM)
*LBF---          BOUNDARY TABLE INDEX FROM BDY NAME              -LBF-

       INTEGER BLANK,BDYNAM
C     BOUNDARY TABLE
C      INDEX- LB=LBDO,LBDE
C      LBNEXT= INCREMENT TO NEXT BOUNDARY
C      LBZ1  = INCREMENT TO THE FIRST BOUNDARY POINT (=0 BEFORE COALLATIO
C      CHNAME= CHANNEL WITH WHICH THE BOUNDARY DATA IS ASSOCIATED
C      UP    = T OR F FOR UPPER OR LOWER BOUNDARY
C      LEDEX = RELATIVE INDEX OF L.E. POINT WHEN LOWER AND UPPER SURFACE
C              CONTOURS ARE CONNECTED
C      BDNAME,LBA,LBB=NAME AND INDEX LIMITS OF SPECIFIC BOUNDARY
C                     DATA WHEN BOUNDARIES ARE COALLATED
       COMMON /CHDATA/ BDT(1),LBNEXT(1),LBZ1(1),
      1                CHNAME(1),UP(1),LEDEX(1),
      2                ZBT(1),RBT(1),ANGBT(42)
         LOGICAL       UP
         INTEGER BDT,CHNAME,BDNAME
         DIMENSION     BDNAME(1),LBA(1),LBB(1)
         EQUIVALENCE   (BDNAME,ZBT), (LBA,RBT), (LBB,ANGBT)
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
         DIMENSION     LIMITS(24)
         EQUIVALENCE   (LIMITS,LHO)
       COMMON /CBITS / BITS,BLANK

C     SEARCH FOR MATCHING BOUNDARY NAME
       LB    = LBDO
   60  IF(BDT(LB).EQ.BLANK .OR. LB.GE.LBDE) GO TO 80
       IF(BDT(LB).EQ.BDYNAM) GO TO 70
       LB    = LB+LBNEXT(LB)
       GO TO 60
   70  LBF   = LB
       RETURN
   80  LBF   = 0
       RETURN
       END
```

```
*DECK LFIT1
      SUBROUTINE LFIT1(X,Y,NPTS, XC,YC,NXC)
*LFIT1          LINEAR FIT INTERPOLATION                        -LFI1-
       DIMENSION          X(10),Y(10), XC(10),YC(10)

C     INPUT-
C      X,Y    = LIST OF COORDINATES DESCRIBING THE INPUT FUNCTION
C      NPTS   = NUMBER OF X,Y POINTS
C      XC     = LIST OF X-S AT WHICH INTERPOLATION IS TO BE PERFORMED
C      NXC    = NUMBER OF XC-VALUES

C     OUTPUT-
C      YC     = LIST OF VALUES INTERPOLATED AT XC(IC),IC=1,NXC

C     NOTES-
C      IF  XC  IS OUTSIDE OF THE RANGE OF  X,  THE END VALUE OF  Y  IS SU
C         FOR YC.
C      X  MUST BE LISTED FROM SMALLEST TO LARGEST.
C      DOUBLE  X-POINTS ARE ALLOWED FOR A FUNCTION DISCONTINUITY.

       N      = NPTS
       I      = 1

C     BEGIN INTERPOLATION LOOP FOR  XC(IC),IC=1,NXC
       IC     = 1
    60 XCIC   = XC(IC)
       IF(N.GT.1) GO TO 100
       YC(IC)=Y(1)
       GO TO 190

   100 XG     = X(I+1)-XCIC
       IF(XG) 114,114,102
   102 XF     = XCIC-X(I)
       IF(XF) 110,120,120

C       F.LT.0. (F IS THE FRACTIONAL POSITION IN THE INTERVAL)
   110 I      = I-1
       IF(I) 100,111,100
   111 I      = 1
       YC(IC)= Y(1)
       GO TO 190

C       F.GE.1.
   114 I      = I+1
       IF(I-N) 100,115,100
   115 I      = N-1
       YC(IC)= Y(N)
       GO TO 190

C     INTERPOLATE
   120 YC(IC)= (Y(I)*XG+Y(I+1)*XF)/(XG+XF)

C     INDEX TO NEXT  XC(IC)
   190 IC     = IC+1
       IF(IC.LE.NXC) GO TO 60

       RETURN
       END
```

```
*DECK LOC2
      FUNCTION LOC2(IA,IB)
CLOC2--- CDC VERSION
C     IABS( ADDRESS(IB)-ADDRESS(IA) )
      LOC2  = IABS( LOCF(IB)-LOCF(IA) )
      RETURN
      END
```

```
*DECK LSPFIT
      SUBROUTINE LSPFIT(X,Y,NPTS, XC,YC,NXC,ND)
*LSPFIT          INTEGRATE OR INTERPOLATE                        -LSPFIT-
C     INTEGRATE OR INTERPOLATE USING A PARABOLA WHICH PASSED THROUGH THE
C     AND (I+1) POINTS BUT MISSES THE (I-1) AND (I+2) POINTS (IF THEY BO
C     EXIST) SUCH THAT THE SQUARE OF THE DEVIATION IS A MINIMUM.  NOTE
C     THAT  I  IS GENERALLY SELECTED SUCH THAT
C               X(I).LE.XC.LT.X(I+1)
C     THE EQUATION FOR THE PARABOLA IS
C               Y-Y(I) = B*(X-X(I)) + C*(X-X(I))**2

      DIMENSION X(10),Y(10), XC(10),YC(10)
C      NOTE. THE DIMENSION -10- DOES NOT NEED TO AGREE WITH THE CALLING

C     INPUT-
C      X, Y     PTS. ON CURVE
C      NPTS     NO. OF X
C      XC       LIST OF X AT WHICH CALC TO BE DONE
C      YC(1)    INTEGRATION CONSTANT IF  ND=-1
C      NXC      NO. OF XC
C      ND       =0 TO GET COORD,  =1 TO GET 1ST DERIVATIVE,
C               =-1 FOR INTEGRATION
C      LEND   = LINEAR FIT IN END INTERVAL, T OR F

C     OUTPUT
C      YC       COORDINATE OR DERIVATIVE AT XC    OR
C               YC(IC)= INTEGRAL(Y*DX) FROM XC(1) TO XC(IC) WHERE IC=2,NXC

C     NOTES-
C      -X- MAY BE IN EITHER ASCENDING OR DESCENDING ORDER.
C      FOR INTEGRATION -XC- MUST BE IN THE SAME ORDER AS -X-.  FOR INTERP
C         NO SPECIAL ORDER IS REQUIRED.

      COMMON /CLSPF / I,LEND
       LOGICAL          LEND

      LOGICAL         WITHIN
      DATA KNAME/6HLSPFIT/

      N      = NPTS-1
      IF(ND.EQ.(-1)) I=1
      ISAVE  = 0
      SGN    = SIGN(1.,X(N+1)-X(1))

C     BEGIN INTERPOLATION LOOP FOR XC(IC)  IC=1,NXC
      IC     = 1

C     LOCATE APPROPRIATE INTERVAL
  100 I      = MAXO(1,MINO(I,N))
      WITHIN= .FALSE.
      NCOUNT= N
  102 IF(NCOUNT) 119,103,103
  103 NCOUNT= NCOUNT-1

      XI     = X(I)
      XD     = XC(IC)-XI
      IF(N) 104,120,104
  104 IF(SGN*XD) 105,107,110
```

```
C       F.LT.0. (F IS THE FRACTIONAL POSITION IN THE INTERVAL)
  105 IF(I.EQ.1)      GO TO 120
      IF(ND.EQ.(-1)) GO TO 119
      I     = I-1
      GO TO 102

C       F.EQ.0
  107 IF(X(I+1).NE.XI) GO TO 120
      IF(I.GE.N) GO TO 105
      GO TO 116

C       F.GT.0.
  110 IF(SGN*(XC(IC)-X(I+1))) 120,112,114

C       F.EQ.1.0,  CHECK FOR INTEGRATION AND DOUBLE POINT BEFORE INCREMEN
  112 IF((ND.EQ.(-1)) .OR. (I.NE.N .AND. X(I+1).EQ.X(I+2))) GO TO 120

C       F.GT.1.0
  114 IF(I.EQ.N) GO TO 120
      IF(ND.EQ.(-1)) GO TO 122
  116 I     = I+1
      GO TO 102

  119 CALL ERRORK(KNAME)

C     PRELIMINARY CALCULATIONS FOR INTERPOLATION OR INTEGRATION
  120 WITHIN=.TRUE.
  122 IF(I-ISAVE) 124,129,124
  124 ISAVE = I
      YI      = Y(I)
      X3      = X(I+1)-XI
      Y3      = Y(I+1)-YI
      C       = 0.
      TOP     = 0.
      BOT     = 0.
      IF(LEND .AND. (I.EQ.1 .OR. I.EQ.N)) GO TO 128
      IF(I.LE.1) GO TO 127
      X1      = X(I-1)-XI
      X13     = X(I-1)-X(I+1)
      TOP     = X1*(Y3*X1-(Y(I-1)-YI)*X3)*X13
      BOT     = X1*X1*X13*X13*X3
  127 IF(I.GE.N .OR. (XD.EQ.0..AND.BOT.NE.0.)) GO TO 128
      X4      = X(I+2)-XI
      X43     = X(I+2)-X(I+1)
      Y4      = Y(I+2)-YI
      TOP     = TOP + X4*(Y3*X4-Y4*X3)*X43
      BOT     = BOT + X4*X4*X43*X43*X3
  128 IF(BOT.NE.0.) C = -TOP/BOT
      B       = 0.
      IF(N.GT.0 .AND. X3.NE.0.) B = (Y(I+1)-YI)/X3 - C*X3
  129 IF(ND) 130,140,141

C     ND=-1, INTEGRATE
  130 IF(.NOT.WITHIN) XD=X3
      S1      = (YI + (B/2. + C/3.*XD)*XD)*XD
      IF(WITHIN) GO TO 135
C     -I- IS BEING INCREMENTED TO FIND APPROPRIATE INTERVAL.  HENCE,
```

```
C       CUMULATE THE INTEGRAL OF THE ITH INTERVAL.
        SA    = SA + S1
        GO TO 116
C     APPROPRIATE INTERVAL FOUND.    X(I)-XC(IC)-X(I+1)
  135 IF(IC.EQ.1) SA=YC(IC)-S1
        IF(IC.NE.1) YC(IC)=SA+S1
        GO TO 150

C     ND=0,  INTERPOLATE FOR COORDINATES
  140 YC(IC)= YI + (B + C*XD)*XD
        GO TO 150

C     ND=1,  FIRST DERIVATIVE
  141 YC(IC)= B + 2.*C*XD
        GO TO 150

  150 IC    = IC+1
        IF(NXC-IC) 900,160,160
  160 IF(ND.NE.(-1).AND.XC(IC).EQ.XC(IC-1)) I=I+1
        GO TO 100

  900 RETURN
        END
```

```
*DECK LSUM
      SUBROUTINE LSUM(X,Y,N, S)
*LSUM--          CUMULATIVE TAPEZOIDAL INTEGRATION                -LSUM-
      DIMENSION X(9),Y(9),S(9)
      DO 90 I=2,N
   90 S(I)  = .5*(Y(I)+Y(I-1))*(X(I)-X(I-1)) +S(I-1)
      RETURN
      END
```

```
*DECK MBEGIN
      FUNCTION MBEGIN(J2)
*MBEGIN          FIND FIRST FIELD POINT                          -MBEGIN-
C                FOR A GIVEN STREAMLINE

C     INPUT
C      J2     = STREAMLINE INDEX
C     OUTPUT-
C      MBEGIN= FIELD INDEX OF FIRST POINT ON THE SL

      COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
     *                LO,LESTA,  LDUM(8),
     *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
     *                LEO,LEE, LRO,LRE,LRD
        DIMENSION    LIMITS(24)
        EQUIVALENCE  (LIMITS,LHO)

      COMMON /CIDEX / M,J,MU,MD,ISTAG
      DATA KNAME/6HMBEGIN/

C     SEARCH FOR FIRST POINT ON STREAMLINE  J
  101 M      = 1
  105 CALL GETIX
      IF (J.EQ.J2 .AND. MU.EQ.0) GO TO 115
  110 IF(M.EQ.NM) CALL ERRORK(KNAME)
  112 M      = M+1
      GO TO 105

  115 MBEGIN= M
      RETURN
      END
```

PRECEDING PAGE BLANK NOT FILMED

```
*DECK MOVE
      SUBROUTINE MOVE(NR,X1,Y1,N1,ND1,X2,Y2,N2,ND2,X3,Y3,N3,ND3)
CMOVE ---- FORTRAN SIMULATION OF MOVE (CDC)
      DIMENSION X1(1),Y1(1),X2(1),Y2(1),X3(1),Y3(1)
      DO 100 L=1,NR
      GO TO (5,10,15) , L
    5 N      = IABS(N1)
      ND     = ND1
      IF( N1.LT.0 ) ND=-1
      NS     = N1
      GO TO 40
   10 N      = IABS(N2)
      ND     = ND2
      IF( N2.LT.0 ) ND=-1
      NS     = N2
      GO TO 40
   15 N      = IABS(N3)
      ND     = ND3
      IF( N3.LT.0 ) ND=-1
      NS     = N3
   40 K      = 1
      IF(NS)401,100,41
  401 K      = N
   41 IF( (K.LE.0) .OR. (K.GT.N) .OR. NS.EQ.0 ) GO TO 100
      GO TO (45,50,55) , L
   45 Y1(K) = X1(K)
      GO TO 80
   50 Y2(K) = X2(K)
      GO TO 80
   55 Y3(K) = X3(K)
   80 K      = K+ND
      GO TO 41
  100 CONTINUE
      RETURN
      END
```

```
*DECK SETM
      SUBROUTINE SETM(NR,VAL,X1,N1,X2,N2,X3,N3)
      DIMENSION X1(1),X2(1),X3(1)
CSETM ---- FORTRAN SIMULATION OF SETM(CDC)
      DO 200 L=1,NR
      GO TO (105,110,115) , L
  105 NS     = N1
      GO TO 140
  110 NS     = N2
      GO TO 140
  115 NS     = N3
  140 DO 180 K=1,NS
      GO TO (145,150,155) , L
  145 X1(K) = VAL
      GO TO 180
  150 X2(K) = VAL
      GO TO 180
  155 X3(K) = VAL
  180 CONTINUE
  200 CONTINUE
      RETURN
      END
```

```
*DECK FMPYC
      SUBROUTINE FMPYC(NR,C,X1,Y1,N1,X2,Y2,N2,X3,Y3,N3)
      DIMENSION X1(1),Y1(1),X2(1),Y2(1),X3(1),Y3(1)
CFMPYC ---- FORTRAN SIMULATION OF FMPYC (CDC)
      DO 300 L=1,NR
      GO TO (205,210,215) , L
  205 NS     = N1
      GO TO 240
  210 NS     = N2
      GO TO 240
  215 NS     = N3
  240 DO 280 K=1,NS
      GO TO (245,250,255) , L
  245 Y1(K) = C*X1(K)
      GO TO 280
  250 Y2(K) = C*X2(K)
      GO TO 280
  255 Y3(K) = C*X3(K)
  280 CONTINUE
  300 CONTINUE
      RETURN
      END
```

```
*DECK QIREM
      SUBROUTINE QIREM(X,Y, XJP, QV)
*QIREM-       QUADRATIC INTERPOLATION ROOT EVALUATION        -QIREM-
C             FOR FUNCTIONS WITH MAXIMUMS

       DIMENSION QV(8)
      DATA KNAME/6HQIREM /

C     INPUT-
C      X      = ABSCISSA
C      Y      = ORDINATE (OR ERROR)
C      XJP    = X-JUMP TO BE TAKEN BEFORE ROOT/MAX IS SPANNED,  THE SIGN I
C               A POSITIVE ERROR
C      QV     = STORAGE FOR EIGHT ELEMENT QIRE VECTOR
C      QV(1)  = CTR =0. (FIRST ENTRY ONLY)
C      YTOL   = TOLERANCE ON THE ERROR
C      YO     = ORDINATE TO BE OBTAINED (OPTIONAL)
C      DYDX   = ESTIMATE OF SLOPE FOR 2ND GUESS (OPTIONAL)
C      CTRMAX= MAXIMUM NO. OF ITERATIONS (=25 IF NOT SPECIFIED)

C     OUTPUT-
C      X      = NEXT X ESTIMATE
C      QV(1)  = 0. IF  YTOL  HAS BEEN SATISFIED
C      QV(5)  = 0. IF MAX PT HAS BEEN FOUND WITHIN YTOL,
C                    AND ABS(E).GT.YTOL.

C     NOTES-
C      C      = THIRD COEFFICIENT IN THE EQUATION- Y=A+B*X+C*X**2
C             = D12 IN QIRE NOTATION
C      N1     = EXIT VALUE OF QV(5),  N1=4 IF  X  IS THE PRECICTED MAX PT,
C               N1=+5(-5) IF X IS JUST TO THE LEFT(RIGHT) OF THE PREVIOUSL
C               PREDICTED MAX PT, N1=6 IF X IS THE SECOND PT CLOSE TO THE
C               OTHERWISE N1=N.
C      M      = ENTRY VALUE OF QV(5)
C      SGM    = SIGN OF  M  IF ABS(M)=5
C      SDYDX  = SIGN OF THE SLOPE OF THE CURVE
C      XJ     = JUMP TO BE TAKEN FROM LAST  X
C      XJA    = ABSOLUTE VALUE OF MAXIMUM JUMP = ABS(XJP)
C      XM     = DISTANCE FROM CENTRAL PT TO MAX/MIN OF PARABOLA, =XMAX-XX(
C               OR = DISTANCE FROM CENTRAL PT TO THE ROOT,  =XROOT-XX(2)
C      X1     = INPUT (OR LAST) X VALUE

      COMMON /CQIREM/ YTOL,YO,DYDX,CTRMAX
      COMMON /ERASE / BOT,C,DXDY,E,I,II,IN,ISPAN,M,N,RADICL,SDYDX,SGN,
     1                TOP,X1,X13,X13P,XJ,XJA,XM, DX(3),DY(3),QV1(10)
       DIMENSION XX(4),YY(4)
       EQUIVALENCE (CTR,QV1(1)), (N1,QIND,QV1(5)),
     1             (XX,QV1(2)), (YY,QV1(6))

C     INITIALIZING  AND  PRELIMINARY CHECKING
      IF(CTRMAX.EQ.0.) CTRMAX=25.
      DO 30 I=1,8
   30  QV1(I)= QV(I)
      N1     = IFIX(QV1(5))
      E      = Y-YO
      M      = N1
      IF(CTR.EQ.0.) M=0
      SGM    = 1.
```

```
      IF(M.GE.0) GO TO 36
      M      = 5
      SGM    = -1.
   36 N      = MIN0(M,3)
C     SDYDX = SIGN(1.,-XJP)
C     (ALTERNATE CALC TO CIRCUMVENT COMPILER ERROR)
      IF(XJP) 41,42,42
   41 SDYDX = 1.
      GO TO 43
   42 SDYDX = -1.
   43 XJA    = ABS(XJP)
      X1     = X
      IF(M-5) 44,45,46
   44 IF(ABS(E).LE.YTOL) GO TO 800
      IF(M.EQ.4 .AND. ABS(E-YY(2)).LE.YTOL) GO TO 700
      IF(CTR.GE.CTRMAX) CALL ERRORK(KNAME)
      GO TO 50
   46 M      = 3
   45 X13P   = XX(3)-XX(1)

C    DETERMINE INDEX FOR INSERTING CURRENT X,E INTO XX,YY TABLE WHICH IS
C     ORDERED ACCORDING TO X.
   50 IN     = 1
      IF(N.EQ.0)        GO TO 90
   60  IF(XX(IN).GT.X1) GO TO 70
       IN     = IN+1
       IF(IN.LE.N)      GO TO 60
      GO TO 90

C    RELOCATE IN PREPARATION FOR INSERTING  X,E
   70 II     = N+1
   80  XX(II)= XX(II-1)
       YY(II)= YY(II-1)
       II     = II-1
       IF(II.NE.IN) GO TO 80

C    INSERT NEW POINT
   90 N      = N+1
      XX(IN)= X1
      YY(IN)= E

C    LOCATE INTERVAL WHICH SPANS ROOT
       ISPAN = 0
       IF(N.EQ.1) GO TO 200
        DO 110 I=2,N
        IF(SDYDX*YY(I).GT.0. .AND. SDYDX*YY(I-1).LT.0.) ISPAN=I
  110   CONTINUE

C    REDUCE  XX,YY TABLE TO THREE POINTS
       IF(N.LE.3)     GO TO 200
       IF(ISPAN.EQ.0) GO TO 140
C     (ROOT HAS BEEN SPANNED)
  122  IF(ISPAN.EQ.N) GO TO 150
       IF(ISPAN.EQ.2) GO TO 175
       IF(ABS(YY(1)).GT.ABS(YY(4))) GO TO 150
       GO TO 175

C     (ROOT HAS NOT BEEN SPANNED)
```

```
  140 IF(IN.LE.2)    GO TO 175

C      DELETE FIRST POINT
  150  DO 160 I=1,N
       XX(I) = XX(I+1)
  160  YY(I) = YY(I+1)
       ISPAN = ISPAN-1
C      DELETE FOURTH POINT
  175 N      = N-1

C     SIMPLE X-JUMP PREDICTION
  200 N1     = N
       IF(ISPAN.GT.0 .OR. DYDX.NE.0.) GO TO 205
C      XJ     = SDYDX*SIGN(XJA,-E)
C      (ALTERNATE CALC TO CIRCUMVENT COMPILER ERROR)
       XJ     = XJP
       IF(E.LT.0.) XJ=-XJ
       GO TO 900

C     CURVE FIT PREDICTIONS
  205 IF(N-2) 210,220,300

C     ONE POINT PREDICTION BASED ON INPUT VALUE OF DXDY
  210 XJ     = -E/DYDX
       GO TO 900

C     TWO POINT STRAIGHT LINE PREDICTION
  220 BOT    = YY(2)-YY(1)
       IF(BOT.EQ.0.) GO TO 230
       DXDY   = (XX(2)-XX(1))/BOT
       IF(DXDY*SDYDX.GT.0.) GO TO 240
C      (CURVE SLOPE IS WRONG - MOVE TOWARD MAXIMUM POINT)
  230 XJ     = -3.*SDYDX*XJA
       GO TO 900
C      (CURVE SLOPE IS CORRECT)
  240 XJ     = -E*DXDY
       GO TO 900

C     PARABOLIC CURVE FIT PREDICTION
  300 DX(1) = XX(1)-XX(2)
       DX(3) = XX(3)-XX(2)
       DY(1) = YY(1)-YY(2)
       DY(3) = YY(3)-YY(2)
       BOT    = DX(1)*DY(3) - DX(3)*DY(1)
       IF(ABS(BOT).LT.1.E-12) GO TO 600
       TOP    = DX(1)*DX(1)*DY(3) - DX(3)*DX(3)*DY(1)
       XM     = .5*TOP/BOT
       X13    = XX(3)-XX(1)
       IF(ABS(XM).GT.ABS(1.E3*X13)) GO TO 600
       C      = BOT/(DX(1)*DX(3)*X13)
       RADICL= XM*XM - YY(2)/C
       IF(RADICL.LE.0.) GO TO 360
       SGN    = SIGN(1.,SDYDX*C)
       XM     = XM + SGN*SQRT(RADICL)
       GO TO 890
C      (IMAGINARY ROOT, HENCE WE ARE LOOKING FOR THE MAXIMUM POINT,
C      PREDICT MAX PT IF M=3, SELECT PTS ON LEFT/RIGHT SIDE OF PREVIOUSLY
C      PREDICTED PT IF M=4/5)
```

```
360 IF(M-4) 363,364,365
363 IF(ABS(XM).LT.XJA) N1=4
    GO TO 890
364 XJ     = -X13/8.
    N1     = 5
    IF(IN.GT.2) GO TO 900
    XJ     = -XJ
    N1     = -5
    GO TO 900
365 XJ     = SGM*X13P/4.
    N1     = 6
    GO TO 900

C     RETREAT TO LINEAR INTERPOLATION
600 IF(ISPAN.GT.0) GO TO 122
    GO TO 140

C     MAXIMUM FOUND
700 QIND   = 0.
    GO TO 930

C     SOLUTION FOUND
800 CTR    = 0.
    GO TO 930

C     FINIS
890 X1     = XX(2)+XM
    GO TO 910
900 X1     = X1+XJ
910 CONTINUE
    X      = AMAX1(XX(1)-XJA,AMIN1(X1,XX(N)+XJA))
    CTR    = CTR+1.
930 DO 950 I=1,8
950 QV(I) = QV1(I)
    QV(5) = FLOAT(N1)
999 RETURN
    END
```

```
*DECK TABPRT
      SUBROUTINE TABPRT(NAME, A,NA,NCOL1)
CTABPRT--- CDC VERSION
      DIMENSION A(10)
C     INPUT-
C      NAME  = ARRAY NAME TO BE PRINTED
C      A     = ARRAY TO BE PRINTED
C      NA    = NUMBER OF ELEMENTS
C      NCOL1 = NUMBER OF COLS. TO BE USED IN PRINT FORMAT
C      $$$$$   (MAXIMUM = NA )
C      IITAB = LOC. OF FIRST ELEMENT IN A TO BE PRINTED
C
      COMMON /CBITS  / IBITS,BLANK
      COMMON /CTABPR/ IITAB
      EQUIVALENCE     (LSPACE,ASPACE) , (IB,B)
      DIMENSION FMT(12)
      REAL  I12
      INTEGER  HOLL,HTEST
      DATA IBCI/0010000000000/
      DATA (FMT(J),J=1,12)/10H(1X,I5    ,10H          ,10H          ,
     *10H          ,10H          ,10H          ,
     *10H          ,10H          ,10H          ,
     *10H          )/
      DATA
     * F1,       F3,       F6,       E5,       BCD,      OCT,      I12/
     *6H,F12.1, 6H,F12.3, 6H,F12.6, 6H,E12.4, 6H,6X,A6, 6H,8X,O4, 4H,I12
     */
      DATA HMASK/0000000000077777777/ , HTEST/0000000000055555555/,
     *     INMASK/0377777770000000000/
      DATA NINMSK/0777700000000000000/

      NCOL  = MINO(NCOL1,10)
      NB    = NA

C     WRITE  HEADING
      WRITE (6,1000) NAME

   45 I1    = IITAB
      I     = I1
      I2    = 0

C     WRITE LINE SPACE
   47 WRITE (6,1002)
C     LOCATION OF NEXT LINE SPACE IS GIVEN BY A(I+1)
      ASPACE= A(I+1)
      IF( LSPACE.LE.1 .OR. LSPACE.GE.IBCI ) LSPACE=IBCI
      LSPACE= LSPACE+I-1
      GO TO 110

C     BEGIN LOOP TO DEFINE LINE FORMAT
   48 II    = 1
C
   50 B     = A(I)

C     SPECIAL NUMBERS                                    46
      NN    = NINMSK.AND.B
      IF( NN.EQ.NINMSK ) GO TO 82
C     TEST FOR HOLLERITH (6H------  MAX.)
```

```
      HOLL    = HMASK.AND.B
      IF( HOLL.EQ.HTEST ) GO TO 80
C     TEST FOR INTEGER  (BITS 36-58=0 FOR MAX 635 INTEGER
C     FLOATING POINT NUMBERS NORMALIZED
      IF( IB.EQ.IBITS ) GO TO 85
      INTGR = INMASK.AND.IB
      IF( INTGR.EQ.0 ) GO TO 82
C     REAL NUMBER -- NORMALIZED
      B       = ABS(B)
      FMT(II+1)= E5
      IF( B.LT.1.E-3 .OR. B.GE.1.E8 ) GO TO 90
   65 FMT(II+1)= F6
      IF( B.GE.1.E3 ) FMT(II+1)=F3
      IF( B.GE.1.E5 ) FMT(II+1)=F1
      GO TO 90

C     BCD
   80 FMT(II+1)= BCD
      GO TO 90

C     INTEGER
   82 FMT(II+1)= I12
      GO TO 90

C     OCTAL
   85 FMT(II+1)= OCT
   90 II      = II+1
      I       = I+1
      IF( I.GT.LSPACE ) GO TO 100
      IF( II.LE.NCOL .AND. I.LE.NB ) GO TO 50
  100 I2      = I-1
      WRITE (6,FMT) II,(A(I),I=II,I2)
      II      = I
  110 IF( I.GE.NB ) GO TO 990
      IF( I.GT.LSPACE ) GO TO 47
      GO TO 48
  990 IITAB = 1
 1000 FORMAT(/2X,A6)
 1002 FORMAT(1H )
      RETURN

      END
```

```
*DECK TAN
      FUNCTION TAN(X)
*TAN---                                                        -TAN-
      TAN    = SIN(X)/COS(X)
      RETURN
      END
```

```
*DECK SS5PT
      SUBROUTINE SS5PT
*SS5PT          SUPERSONIC 5-PT FORMULA                          -SS5PT-

C     INPUT-
C      X(1-4)= POINT SPACING FROM POINT ZERO
C      A4FACT= 1 FOR CUBIC, =0 FOR SAME A4 AS A PARABOLA

C     OUTPUT-
C      A0,A1,A2,A3,A4= INFLUENCE COEFFICIENTS FOR D2Y/DX2 AT X(4)

       COMMON /CSS    / SSFML,SSEF,SSEANG,SSDF,SSFEND,SSFND1
      1                 ,SSDLE,A4FACT,BRLX,CURRLX,TSIC
        INTEGER          SSFML
        LOGICAL                SSEF,        SSDF,    SSDLE

       COMMON /CSS5PT/ X(4),Y(4), X21,X31,X32,X41,X42,X43, A0,A1,A2,A3,A4

       X43    = X(4)-X(3)
       X42    = X(4)-X(2)
       X41    = X(4)-X(1)
       X32    = X(3)-X(2)
       X31    = X(3)-X(1)
       X21    = X(2)-X(1)

       A4     = 2./(X42*X43)*(1.+A4FACT*(X42+X43)/X41)
       A1     = (-A4*X(4)*X42*X43 + 2.*(X(4)+X42+X43))/
      1          (X(1)*X21*X31)
       A2     = (+A4*X(4)*X41*X43 - 2.*(X(4)+X41+X43))/
      1          (X(2)*X21*X32)
       A3     = (-A4*X(4)*X41*X42 + 2.*(X(4)+X41+X42))/
      1          (X(3)*X31*X32)
       A0     = -(A1+A2+A3+A4)

       RETURN
       END
```

```
*DECK STANO
      SUBROUTINE STANO(M,LR,UPPER)
*STANO-          STATION INDEX FROM FIELD POINT                  -STANO-
       LOGICAL                 UPPER

C     INPUT-
C      M      = FIELD PT INDEX
C      LR     = 0 FOR FIRST ENTRY OTHERWISE LR.NE.0
C     OUTPUT-
C      LR     = STATION TABLE INDEX
C      UPPER  = T IF  M  IS AN UPPER BOUNDARY POINT, =F OTHERWISE

       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
         DIMENSION    LIMITS(24)
         EQUIVALENCE  (LIMITS,LHO)
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
       COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
      1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
      1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
      3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
         LOGICAL      PRIM
         INTEGER TYPELB,TYPEUB
         DIMENSION    SCHOKE(1)
         EQUIVALENCE  (SCHOKE,DWDV)
       DATA KNAME/6HSTANO /

       L      = LR
       IF(L.EQ.0) L = LO
       UPPER = .FALSE.
       LSAV   = L
       LSTOP = 999999

  120  IF(L.GE.LSTOP) CALL ERRORK(KNAME)
       IF(MUB(L).EQ.M) GO TO 150
       IF(M.GE.MLB(L) .AND. M.LE.MUB(L)) GO TO 160
       L      = L+LNEXT(L)
       IF(L.LT.LESTA) GO TO 120
       L      = LO
       LSTOP = LSAV
       GO TO 120

  150  UPPER = .TRUE.
  160  LR      = L

       RETURN
       END
```

```
*DECK STAX1
      SUBROUTINE STAX1(X1FIND,X2B,X2A,LXB,LXA)
*STAX1-         STATION INDEX FROM X1 AND X2-COORDINATES     -STAX1-

C     INPUT-
C      X1FIND= X1-COORDINATE
C      X2B    = X2-COORDINATE OF UPPER BOUNDARY (I.E. STATION BELOW THE BO
C      X2A    = X2-COORDINATE OF LOWER BOUNDARY (I.E. STATION ABOVE THE BO

C     OUTPUT-
C      LXB    = INDEX OF STATION WHICH CONTAINS COORDINATES-X1FIND,X2B
C      LXA    = INDEX OF STATION WHICH CONTAINS COORDINATES-X1FIND,X2A

       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
        DIMENSION      LIMITS(24)
        EQUIVALENCE    (LIMITS,LHO)
       COMMON /SLTAB / W(128),X2(128),SLCHN(128)
        INTEGER SLCHN
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
       COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
      1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
      1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
      3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
        LOGICAL        PRIM
        INTEGER TYPELB,TYPEUB
        DIMENSION      SCHOKE(1)
        EQUIVALENCE    (SCHOKE,DWDV)

       COMMON /CIDEX / M,J,MU,MD,ISTAG
       DATA KNAME/6HSTAX1 /

       NFOUND= 0
       IF(X2B.GE.0.) NFOUND=1
       IF(X2A.GE.0.) NFOUND=NFOUND+1
       L      = LO

  110  IF(X1(L).NE.X1FIND) GO TO 120
       M      = MUB(L)
       CALL GETIX
       IF(X2(J).NE.X2B) GO TO 115
       LXB    = L
       NFOUND= NFOUND-1
       GO TO 120
  115  M      = MLB(L)
       CALL GETIX
       IF(X2(J).NE.X2A) GO TO 120
       LXA    = L
       NFOUND= NFOUND-1
  120  L      = L+LNEXT(L)
       IF(NFOUND.EQ.0) GO TO 130
       IF(L.LT.LESTA) GO TO 110
```

51

```
*DECK STCN
      PROGRAM STCN
      COMMON /CTAPOS/  RESTRT,ENDBDT,STCFIL,K6SV
      LOGICAL          RESTRT,ENDBDT,STCFIL
      COMMON /SELECT/ LENTRY
    1 GO TO  (5,10) , LENTRY
C     READ INPUT
    5 CALL OVERLAY(3HSTC,1,1,6HRECALL)
      GO TO 20
C     BUILD TABLES
   10 IF(RESTRT) GO TO 15
      LENTRY= 1
      CALL OVERLAY(3HSTC,1,2,6HRECALL)
      CALL OVERLAY(3HSTC,1,3,6HRECALL)
      LENTRY= 2
   12 CALL OVERLAY(3HSTC,1,2,6HRECALL)
      GO TO 20
C  RESTRT CASE
   15 LENTRY= 2
      CALL OVERLAY(3HSTC,1,3,6HRECALL)
      LENTRY= 3
      GO TO 12
   20 RETURN
      END
```

```
*DECK ERRORN
      SUBROUTINE ERROR1
CEDUMPN         STC EDUMP - INPUT LINK                          -EDUMPN-

       COMMON /ALLCOM/ MACHA,PSA,TSA,PTA,TTA, AXIA,RGA,GAMA,
      1                MACHC,PSC,TSC,PTC,TTC, AXIC,RGC,GAMC,
      2                DAXIT,SCALEA,TTE,CHOTST
        REAL           MACHA(1),MACHC
        LOGICAL        AXIA,AXIC
        LOGICAL        CHOTST
C     CHANNEL INPUT DATA TABLE
C      INDEX- LH=LHO,LHE
       COMMON /CHDATA/ CHNAM(1),LHNEXT(1),WTFLOW(1),TTO(1),PTO(1),
      1                TSO(1),PSO(1),MACHO(1),AO(1),VARY(1),
      2                RG(1),GAM(1), NR(1),NC(1),TAB(6),
      4                BB(75)
        LOGICAL        VARY
        INTEGER CHNAM
        DIMENSION      VO(1)
        REAL           MACHO
        EQUIVALENCE    (VO,MACHO)
C     BOUNDARY TABLE
C      INDEX- LB=LBDO,LBDE
C      LBNEXT= INCREMENT TO NEXT BOUNDARY
C      LBZ1   = INCREMENT TO THE FIRST BOUNDARY POINT (=0 BEFORE COALLATIO
C      CHNAME= CHANNEL WITH WHICH THE BOUNDARY DATA IS ASSOCIATED
C      UP     = T OR F FOR UPPER OR LOWER BOUNDARY
C      LEDEX = RELATIVE INDEX OF L.E. POINT WHEN LOWER AND UPPER SURFACE
C              CONTOURS ARE CONNECTED
C      BDNAME,LBA,LBB=NAME AND INDEX LIMITS OF SPECIFIC BOUNDARY
C                     DATA WHEN BOUNDARIES ARE COALLATED
       DIMENSION       BDT(1),LBNEXT(1),LBZ1(1),
      1                CHNAME(1),UP(1),LEDEX(1),
      2                ZBT(1),RBT(1),ANGBT(42)
        LOGICAL        UP
        INTEGER BDT,CHNAME,BDNAME
        DIMENSION      BDNAME(1),LBA(1),LBB(1)
        EQUIVALENCE    (BDNAME,ZBT), (LBA,RBT), (LBB,ANGBT)
C     FLOW ADJUSTMENT TABLE
C      INDEX- LF=LFO,LFE
C      NFCOLS= 8
C      X1F    = ORTHOGONAL COORDINATE
C      X2F    = STREAMLINE COORDINATE OF SL EMINATING FROM T.E.
C      X1BF   = X1-COORDINATE OF CHOKE STATION OF FLOW BELOW T.E.
C      X1AF   = X1-COORDINATE OF CHOKE STATION OF FLOW ABOVE T.E.
C      S1F    = S1-COORDINATE OF T.E. (UPPER SURFACE). THIS ITEM
C               IS USED WHEN INTERPOLATING FOR WAKE DELTA-STAR.
C      LFB,LFA=INDICES OF STATIONS BELOW AND ABOVE T.E.
C      NCHB,NCHA=NUMBER OF CHANNELS BELOW AND ABOVE T.E.
C      LRF    = INDEX OF DUMMY ORTCHN LIST FOR THE T.E.
C      LRXF   = INDEX OF LAST CHANNEL BELOW THE T.E.
C      JORDER= 0 IF TOTAL FLOW AT X1F IS GIVEN
C            = 2 IF FLOW ABOVE T.E. IS GIVEN
C            = 1 IF FLOW BELOW T.E. IS GIVEN
C      JORDER= -1 IF FLOW AT X1F IS CHOKED AND SINGLE CHANNEL
       DIMENSION       X1F(1),X2F(1),X1BF(1),X1AF(1),
      1                S1F(1),NCHB(1),NCHA(1),JORDER(1),VNR(12)
        EQUIVALENCE    (LFB,X1BF),(LFA,X1AF),(LRF,NCHB),(LRXF,NCHA)
```

```
      DIMENSION      LFB(1),LFA(1),LRF(1),LRXF(1)
C     TABLE OF CONVECTED PROPERTIES
C      INDEX- LT=LTO,LTE
C      CH     = CHANNELNAME
C      LTNEXT= INDEX INCREMENT TO THE NEXT CHANNEL
C      LPSI   = RELATIVE LOCATION OF PSI LIST
C      NPT    = NO. OF PSI, TT, PT AND RCU VALUES
C      LTT    = RELATIVE LOCATION OF TT LIST
C      LPT    = RELATIVE LOCATION OF PT LIST
C      LRCU   = RELATIVE LOCATION OF RCU LIST
       DIMENSION        CH(1),LTNEXT(1),NPT(1),LPSI(1),LTT(1),LPT(1),
     1                  LRCU(1),
     2                  CRG(1),CPGJ(1),C2CP(1),QGAM(1),FGT(1),FGP(1),
     3                  FGR(1),AREATB(485)
        INTEGER CH
       DIMENSION XCH(1)
       EQUIVALENCE (CH,XCH)
C     STREAMLINE TABLE
       COMMON /SLTAB / W(128),X2(128),SLCHN(128)
        INTEGER SLCHN
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
       DIMENSION        X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
     1                  TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
     1                  TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
     3                  VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
        LOGICAL         PRIM
        INTEGER TYPELB,TYPEUB
        DIMENSION       SCHOKE(1)
        EQUIVALENCE     (SCHOKE,DWDV)
C     TABLE OF WAKE DISPLACEMENT THICKNESS
C      INDEX- LW=LWO,LWE
       DIMENSION                X2W(1),LWNEXT(1),S1W(47)
        DIMENSION       DST(1)
        EQUIVALENCE     (DST,S1W)
C      SUBTABLE ARRANGEMENT IS-
C       X2W,LWNEXT(=2+2N), S1W(1),S1W(2)...S1W(N), DST(1),DST(2)...DST(N)
C      X2W    = STREAMLINE COORDINATE
C      S1W    = DISTANCE ALONG STREAMLINE FROM T.E.
C      DST    = WAKE DISPLACEMENT THICKNESS AS A FUNCTION OF S1W

C     FIELD TABLES
C      INDEX- M=MO,NM
       COMMON /CZ     / Z(300)
       COMMON /CR     / R(300)
       COMMON /CS2    / S2(300)
       COMMON /CS1    / S1(300)
       COMMON /CPHI1  / PHI1(300)
       COMMON /CM     / JMS(300)
       COMMON /CCURV  / CURV(300)
       COMMON /CB     / B(300)
       COMMON /CIDEX  / M,J,MU,MD,ISTAG

C     TABLE OF INDEX LIMITS
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
```

```
     *                  LO,LESTA,  LDUM(8),
     *                  MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
     *                  LEO,LEE, LRO,LRE,LRD
       DIMENSION        LIMITS(24)
C     TABLE OF LEADING EDGE AND TRAILING EDGE POINTS
C      INDEX- LE=LEO,LEE,10
C      NLE,NTE=NO. OF L.E. AND T.E. COINCIDENT PTS, RESPECTIVELY
C      CHL,CHU=NAME OF CHANNEL ABOVE AND BELOW PT, RESPECTIVELY
C      BDL,BDU=BOUNDARY NAMES ASSOCIATED WITH THE POINTS
C      NUSED = COUNT OF TIMES THAT POINT USED IN CONSTRUCTION OF /ORTCHN/
       COMMON /LETEPT/ XE(1),YE(1),ANGE(1),NLE(1),NTE(1),
     1                 CHL(1),CHU(1),BDL(1),BDU(1),NUSED(491)
       INTEGER         CHL,CHU,BDL,BDU
C     TABLE OF CHANNELS EMBRACED BY EACH ORTHOGONAL
C      INDEX- LR=LRO,LRE,LRD
C      LRD    = NUMBER OF CHANNELS PLUS ONE,   LR INDEX INCREMENT
C      LEDGE = INDEX OF THE ORTHOGONAL POINT IN THE LETEPT-TABLE
C      LRPREV= POINTER OF LINE OF UPSTREAM CHANNELS IN ORTCHN-TABLE
C      CHNA   = CHANNEL NAMES
       COMMON /ORTCHN/ LEDGE(1),LRPREV(1),CHNA(479)
        INTEGER CHNA
        DIMENSION       JCHNA(1)
        EQUIVALENCE     (JCHNA,CHNA)

        EQUIVALENCE    (CHNAM,BDT,CH,X2W,X1F,X1)
        EQUIVALENCE    (LHNEXT,LBNEXT,LTNEXT,LWNEXT,X2F,LNEXT)
        EQUIVALENCE    (WTFLOW,LBZ1,NPT,S1W,X1BF,MLB)
        EQUIVALENCE    (TTO,CHNAME,LPSI,X1AF,MUB), (PTO,UP,LTT,S1F,PRIM)
        EQUIVALENCE    (TSO,LEDEX,LPT,NCHB,TYPELB)
        EQUIVALENCE    (PSO,ZBT,LRCU,NCHA,NAMELB)
        EQUIVALENCE    (MACHO,RBT,CRG,JORDER,ILB), (AO,ANGBT,CPGJ,VNR,FLB)
        EQUIVALENCE    (VARY,C2CP,S1LB), (RG,QGAM,TYPEUB)
        EQUIVALENCE    (GAM,FGT,NAMEUB), (NR,FGP,IUB), (NC,FGR,FUB)
        EQUIVALENCE    (TAB(1),AREATB,S1UB), (TAB(2),VMB), (TAB(3),DWDV)
        EQUIVALENCE    (TAB(4),X2CL), (TAB(5),VCL), (TAB(6),MCL)

       COMMON /CBITS / BITS,BLANK
       COMMON /CREDIN/ ZTRANS,RTRANS,ROTATE,ZPIVOT,RPIVOT,SCALE,NB,TBB(9)
        EQUIVALENCE     (XTRANS,ZTRANS),(YTRANS,RTRANS),(XPIVOT,ZPIVOT),
     1                  (YPIVOT,RPIVOT)
       COMMON /CTABPR/ I1TAB

       COMMON /CSEGME/ IA(10),IB(10),IMA(10),IMB(10),JTYPE(10),
     1                 N,NSEG, NI,NIM
       COMMON /CSMOOB/ XA(100),YA(100),DEVI(100)
       COMMON /CSMOOC/ DUM1(200),ANG(100),DUM2(400),DEV(100),CURVB(100)
       DATA TXA/2HXA/,TZA/2HZA/

       IGGO   = 1
       GO TO 1777
       ENTRY EDUMP
       IGGO   = 2
 1777  CONTINUE
 1100  FORMAT(///1X36HCHANNEL INPUT DATA TABLE, /CHDATA/ -)
       WRITE (6,1100)
       I1TAB = LHO
       NCX    = NC
       IF(NCX.LT.3) NCX=5
```

```
      CALL TABPRT(BLANK,CHNAM,LHE,NCX)

 1120 FORMAT(///1X54HBOUNDARY COORDINATES AND ANGLES IN RADIANS, /BDYTAB
     */ -)
      WRITE (6,1120)
      I1TAB = LBDO
      CALL TABPRT(BLANK,BDT,LBDE,3)

 1110 FORMAT(///1X41HTABLE OF CONVECTED PROPERTIES, /CONVTB/ -)
      WRITE (6,1110)
      I1TAB = LTO
      CALL TABPRT(BLANK,CH,LTE,7)

      IF(LEE.LT.LEO) GO TO 140
 1130 FORMAT(///1X125HORDERED LIST OF UPSTREAM BOUNDARY PNTS, L.E. PNTS,
     * T.E. PNTS, AND DOWNSTREAM PNTS WITH REFERENCES TO CHANNELS AND BO
     *UNDARIES./1X10H/LETEPT/ -//4X2HLE6X,2HXE10X,15HYE           ANGE12X,
     *3HNLE9X,12HNTE       CHL9X,3HCHU9X,3HBDL9X,3HBDU10X,5HNUSED)
      WRITE (6,1130)
      I1TAB = LEO
      CALL TABPRT(BLANK,XE,LEE,10)

  140 IF(LRE.LT.LRO) GO TO 150
 1140 FORMAT(///1X98HTABULATION OF CHANNELS EMBRACED BY THE ORTHOGONALS
     *WHICH PASS THROUGH THE ABOVE POINTS, /ORTCHN/ -//4X26HLR
     *LE      LR-PREV)
      WRITE (6,1140)
      I1TAB = LRO
      CALL TABPRT(BLANK,LEDGE,LRE,LRO)

 1150 FORMAT(///1X17HSTREAMLINE TABLE-/17X32HJ       X2            SLCHN
     *    W/(I18,F12.6,6X,A6,F12.6,),)
  150 WRITE (6,1150) (J,X2(J),SLCHN(J),W(J),J=1,NJ)

 1190 FORMAT(///1X37HWAKE DISPLACEMENT THICKNESS, /WAKETB///11X19HX2W/S1
     *W          DST)
      WRITE (6,1190)
      I1TAB = LWO
      CALL TABPRT(BLANK,X2W,LWE,2)

 1180 FORMAT(///1X43HTABLE OF FLOW ADJUSTMENT STATIONS, /CADJWF///15X3HX
     *1F9X,3HX2F8X,4HX1RF8X,4HX1AF9X,3HS1F8X,4HNCHB8X,16HNCHA       JORDE
     *R)
      WRITE (6,1180)
      I1TAB = LFO
      CALL TABPRT(BLANK,X1F,LFE,NFCOLS)

 1160 FORMAT(///1X25HSTATION TABLE, /STATAB/ -)
      WRITE (6,1160)
      I1TAB = LO
      CALL TABPRT(BLANK,X1,LESTA,5)

      CALL JMSPRT

 1170 FORMAT(///1X19HFIELD COORDINATES -)
      WRITE (6,1170)
      CALL TABPRT(1HZ,Z,NM,10)
      CALL TABPRT(1HR,R,NM,10)
```

```
C     PRINT OVERALL DATA
      CALL TABPRT(6HALLCOM,MACHA,20,8)


      IF( IGGO.EQ.2 ) RETURN
      LSTOP = 5
      GO TO (900,1777) , LSTOP
  900 RETURN
      END
```

```
*DECK DBSRT1
      SUBROUTINE DBSRT1( F,M,INTR1,INTR2,A,N,II )
*DBSRT1
C   DATE OF THIS VERSION - SEPTEMBER 20,1965
C   SINGLE PRECISION DOUBLE BACK SUBSTITUTION SUBROUTINE USED WITH
C     LRMDS1 SUBROUTINE TO SOLVE SIMULTANEOUS EQUATIONS
      DIMENSION F(II,1),A(II,II),INTR1(1)
      NN=N
      NM1=NN-1
      MM=M
      IF(INTR1(1)) 10,140,10
   10 IF(NN.LE.1) GO TO 40
      DO 30 K=1,NM1
      I1=INTR1(K+1)
      IF(I1) 15,30,15
   15 DO 20 J=1,MM
      X=F(K,J)
      F(K,J)=F(I1,J)
   20 F(I1,J)=X
   30 CONTINUE
   40 DO 90 J=1,MM
      DO 80 L=1,NN
      IF(F(L,J)) 50,80,50
   50 F(L,J)=F(L,J)/A(L,L)
      IF(L.EQ.NN) GO TO 80
      DO 70 I=L,NM1
      IF(A(I+1,L)) 60,70,60
   60 F(I+1,J)=F(I+1,J)-A(I+1,L)*F(L,J)
   70 CONTINUE
   80 CONTINUE
   90 CONTINUE
      IF(NN.LE.1) GO TO 140
  100 DO 130 J=1,MM
      IF(F(NM1+1,J)) 110,130,110
  110 DO 120 I=1,NM1
  120 F(I,J)=F(I,J)-A(I,NM1+1)*F(NM1+1,J)
  130 CONTINUE
      NM1=NM1-1
      IF(NM1) 100,140,100
  140 RETURN
      END
```

```
*DECK ISORT
      SUBROUTINE ISORT(X,Y,Z,B,LB,KGO)
CISORT--- CDC VERSION --MOVE COLUMN DATA TO ARRAYS
      COMMON /CBITS / BITS,BLANK
      DIMENSION X(1),Y(1),Z(1), B(1)
C     INPUT-
C      X,Y,Z       = NEW COLUMNS OF DATA
C      B           = LOCATION OF COLUMN DATA TO BE RELOCATED
C      LB          = 3*COLUMN LENGTH

      K      = 1
      I      = 1
      GO TO ( 10,30 ) , KGO
   10 IF( B(I).EQ.BITS ) GO TO 20
      X(K)   = B(I)
      Y(K)   = B(I+1)
      Z(K)   = B(I+2)
   20 I      = I+3
      K      = K+1
      IF( I.LT.LB ) GO TO 10
      GO TO 50

   30 IF(B(I).EQ.BITS) GO TO 40
      X(K)   = B(I)
      Y(K)   = B(I+1)
   40 I      = I+2
      K      = K+1
      IF(I.LT.LB) GO TO 30
   50 RETURN
      END
```

```
*DECK LOOP
      SUBROUTINE LOOP(A,B,C,N)
*LOOP
C   THIS SUBROUTINE IS USED BY SUBROUTINE LRMDS1
      DIMENSION A(1),B(1)
      DO 10 I=1,N
   10 A(I)=A(I)+B(I)*C
      RETURN
      END
```

```
*DECK LRMDS1
      SUBROUTINE LRMDS1(A,N,INTR1,INTR2,DET,IFACTR,III)
*LRMDS1
C   DATE OF THIS VERSION -- SEPTEMBER 20,1965
C   SINGLE PRECISION LEFT RIGHT MATRIX DECOMPOSITION SUBROUTINE
C   DETERMINANT = DET*(2.0**IFACTR)
C     WHERE (.5) LESS THAN (ABS(DET)) LESS THAN OR EQUAL (1.0)
      DIMENSION A(1),INTR1(1)
      IDIM=III
      NN=N
      NBASE=(NN-1)*IDIM
      NTR=1
      IF(NN.LE.1) GO TO 30
      DO 25 K=2,NN
      INTR1(K)=0
      D=0.0
      M=K
      KM1=K-1
      L=KM1
      JSTOP=KM1+NBASE
      KBASE=(KM1-1)*IDIM
      KKM1=K+KBASE
      KK=KM1+KBASE
      ISTOP=NN+KBASE
      DO 6 I=KK,ISTOP
      B=A(I)
      IKBASE=I-KBASE
*
* MODIFICATION TO SELECT  THE PIVOT ELEMENT AS 1.0 IF PRESENT...
*
* DAVE FERGUSON       10/18/66
*
      IF(B.NE.1.) GO TO 70
      D=1.
      L=IKBASE
      M=IKBASE
      GO TO 80
   70 CONTINUE
*
*
      IF(ABS(B).LE.ABS(D) ) GO TO 3
      D=B
      L=IKBASE
    3 IF(B)4,6,4
    4 M=IKBASE
    6 CONTINUE
   80 CONTINUE
      KM=K-M
      KSTOP=M-KM1
      IF(D) 8,7,8
    7 NTR=0
      INTR2=KM1
      GO TO 60
    8 LKM1=L-KM1
      IF(LKM1) 10,17,10
   10 DO 11 J=KM1,JSTOP,IDIM
      LJ=J+LKM1
      X=A(J)
```

62

```
      A(J)=A(LJ)
   11 A(LJ)=X
      INTR1(K)=L
      NTR=-NTR
   17 KK=KK+IDIM
      DO 22 I=KK,JSTOP,IDIM
      IF(A(I)) 19,22,19
   19 A(I)=A(I)/D
      IF(KM) 20,20,22
   20 Q=-A(I)
      CALL LOOP(A(I+1),A(KKM1),Q,KSTOP)
   22 CONTINUE
   25 CONTINUE
   30 D=0.0
      KM1=NN
      KSTOP=NN+NBASE
      IF(A(KSTOP)) 40,7,40
   40 IFAC=0
      D=1.0
      IDIM1=IDIM+1
      DO 55 K=1,KSTOP,IDIM1
      IF(ABS(A(K)).GE.1.0) GO TO 51
      D=D*2.0
      IFAC=IFAC-1
   51 D=D*A(K)
   52 IF(ABS(D)-1.0) 53,55,54
   53 D=D*2.0
      IFAC=IFAC-1
      GO TO 52
   54 D=D/2.0
      IFAC=IFAC+1
      IF(ABS(D).GT.1.0) GO TO 54
   55 CONTINUE
      IFACTR=IFAC
      IF(NTR.EQ.1) GO TO 60
      D=-D
   60 DET=D
      INTR1(1)=NTR
      RETURN
      END
```

```
*DECK REDBLK
      BLOCK DATA REDBLK
*REDBLK         REDINP BLOCK DATA                                    -REDBLK-
      COMMON /SPACER/ MAXLH,MAXLT,MAXLF,MAXLW
      COMMON /CLWOSV/ LWOSV
      COMMON /CTAPOS/  RESTRT,ENDBDT,STCFIL,K6SV
      LOGICAL          RESTRT,ENDBDT,STCFIL
      DATA MAXLH,MAXLT,MAXLF,MAXLW/400,200,200,200/
      END
      OVERLAY(STC,1,1)
```

```
*DECK STCNR
      PROGRAM STCNR
      CALL REDINP
      RETURN
      END
```

```
*DECK BFACES
      SUBROUTINE BFACES(X,Y,ANG,CURV,E,S,KA,KB)
*BFACES          BEAM FIT EVALUATION OF ANGLE, CURVATURE,    -BFACES-
C                E  AND  S
      DIMENSION X(10),Y(10),ANG(10),CURV(10),E(10),S(10)

C     INPUT-
C      X,Y    - COORDINATES
C      ANG    - ANGLE IN RADIANS  (IF MA=1)
C      ANG(1)= ESTIMATED ANGLE AT THE FIRST POINT (MA=0)
C      KA,KB  - FIRST AND LAST INDEX OF VARIABLES  X,Y,ANG,CURV,E AND S
C      KD     - STORAGE INCREMENT OF  X,Y,ANG,CURV,E, AND S
C      KORDER= 0 IF ERROR1 IS TO BE CALLED WHEN PTS ARE OUT OF ORDER
C            = NON ZERO IF RETURN IS TO BE MADE FOR CORRECTIVE ACTION

C     OUTPUT-
C      ANG    - ANGLE IN RADIANS
C      CURV   - CURVATURE
C      E      - APPLIED FORCES = F/EI  (UNITS ARE 1./L**2)
C      S      - ARC LENGTH ALONG THE CURVE,  (L)
C      KORDER= INDEX OF 2ND OF ADJACENT OUT-OF-ORDER PTS, NOT=0 ON ENTRY

      COMMON /CBEAM / MA,MB,KD
      COMMON /ERASE / A(3),B(1),YPB(1),DA(1),ACHD(1),CHD(793)

      NK     = KB

      CALL BEAM(X(KA),Y(KA),ANG(KA),(KB-KA+KD)/KD)
      IF(KORDER.NE.0) GO TO 800

C     (K=KA)
      I      = 1
      K      = KA
      SK     = S(K)
      E(K)   = 6.*(B(I)+YPB(I))/(CHD(I)*CHD(I))
C     (K=KA,KB-1)
   60 CURV(K)= (4.*B(I)+2.*YPB(I))/(CHD(I)*(1.+1.5*B(I)*B(I)))
      IF(KA-K) 65,80,80
C     (K=KA+1,KB-1)
   65 E(K)   = 6.*((B(I)+YPB(I))/(CHD(I)*CHD(I))
     1           - (B(I-8)+YPB(I-8))/(CHD(I-8)*CHD(I-8)))
C     (K=KA+1,KB)
   70 SK     = SK + CHD(I-8)*(1.+(B(I-8)*B(I-8)-.5*B(I-8)*YPB(I-8)+
     1          YPB(I-8)*YPB(I-8))/15.)
      S(K)   = SK
      IF(K-NK) 80,90,90
   80 I      = I+8
      K      = K+KD
      IF(K-NK) 60,70,70

C     (K=KB)
   90 CURV(K)=(-2.*B(I-8)-4.*YPB(I-8))/(CHD(I-8)*(1.+1.5*YPB(I-8)*YPB(I-
     1          8)))
      E(K)   = -6.*(B(I-8)+YPB(I-8))/(CHD(I-8)*CHD(I-8))
      GO TO 900

C     OUT OF ORDER POINTS
  800 KORDER= KA+KORDER-KD
```

```
      700 RETURN
          END
```

```
*DECK ELLIP
      SUBROUTINE ELLIP(X1,Y1,ANG1,X2,Y2,ANG2,ALPHAD)
*ELLIP          ELLIP AND OTHER SMOOTH DUMMY SUBROUTINES
C   SUBROUTINE TO FIT AN ELLIPSE GIVEN TWO POINTS AND THE ORIENTATION

      ENTRY ELLIPT
C   SUBROUTINE TO FIT AN ELLIPSE WHOSE ORIGIN AND DIMENSION ARE GIVEN IN
C   A ROTATED AND TRANSLATED COORDINATE SYSTEM

      ENTRY XTRUNC
C   FUNCTION TO TRUNCATE XX TO AN EVEN MULTIPLE OF DX

      ENTRY ATDMR
C   SUBROUTINE FOR AUGMENTED TRIDIAGONAL MATRIX REDUCTION

      ENTRY BAD
C   SUBROUTINE TO DELETE BAD DATA BY ADJUSTING DATA LISTS


      ENTRY CUBER
C   SUBROUTINE TO CALCULATE YPP IN TERMS OF Y FOR CUBIC SPLINE EQUATIONS
C   WITH ARBITRARY END CONDITIONS

      ENTRY SMULTI
C   SUBROUTINE TO MULTIPLY TRIADIAGONAL AND SQUARE MATRIX

      ENTRY HYPTS
      ENTRY HYPER1
      ENTRY HYPER2
      RETURN
      END
```

```
*DECK RBD
      SUBROUTINE RBD
*RBD---        READ IN BOUNDARY DATA                                -RBD-

C     INPUT-
C      ENDBDT= END OF BDY/STC TAPE RECORDS, T OR F
C      ENDCRD= END OF ALL STC CARD INPUT, T OR F
C      K6SV  = VALUE OF KEY(6) OF LAST RECORD READ FROM TAPE
C      RESTRT= RESTART (WITH EXISTING TABLES) IS TRUE ONLY
C              IF CARD BDY-DATA HAS NOT YET BEEN ENCOUNTERED
C      STCFIL= T IF A STC-SUBFILE EXISTS ON TAPE=ORGF.
C     OUTPUT-
C      ENDBDT=
C      K6SV  =
C      RESTRT=

       INTEGER          REFS,BDY,CHN
       COMMON /BCOMMN/ PROGM(8),PROGSV,FILIN,FILOT,REFS(5)
        LOGICAL                        FILIN,FILOT

       COMMON /ALLCOM/ MACHA,PSA,TSA,PTA,TTA, AXIA,RGA,GAMA,
      1                MACHC,PSC,TSC,PTC,TTC, AXIC,RGC,GAMC,
      2                DAXIT,SCALEA,TTE,CHOTST
        REAL           MACHA(1),MACHC
        LOGICAL        AXIA,AXIC,CHOTST
C     BOUNDARY TABLE
C      INDEX- LB=LBDO,LBDE
C      LBNEXT= INCREMENT TO NEXT BOUNDARY
C      LBZ1  = INCREMENT TO THE FIRST BOUNDARY POINT (=0 BEFORE COALLATIO
C      CHNAME= CHANNEL WITH WHICH THE BOUNDARY DATA IS ASSOCIATED
C      UP    = T OR F FOR UPPER OR LOWER BOUNDARY
C      LEDEX = RELATIVE INDEX OF L.E. POINT WHEN LOWER AND UPPER SURFACE
C              CONTOURS ARE CONNECTED
C      BDNAME,LBA,LBB=NAME AND INDEX LIMITS OF SPECIFIC BOUNDARY
C                     DATA WHEN BOUNDARIES ARE COALLATED
       COMMON /CHDATA/ BDT(1),LBNEXT(1),LBZ1(1),
      1                CHNAME(1),UP(1),LEDEX(1),
      2                ZBT(1),RBT(1),ANGBT(42)
        LOGICAL        UP
        INTEGER BDT,CHNAME,BDNAME
        DIMENSION      BDNAME(1),LBA(1),LBB(1)
        EQUIVALENCE    (BDNAME,ZBT), (LBA,RBT), (LBB,ANGBT)
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
        DIMENSION      LIMITS(24)
        EQUIVALENCE    (LIMITS,LHO)

       COMMON /ADAM02/ ENDJOB,NUMPLT,PLOTED,ENDCRD
        LOGICAL        ENDJOB,        PLOTED,ENDCRD
       COMMON /CBITS / BITS,BLANK
       COMMON /CLINES/ LINES,OMITFK,PTITLE(6)
        LOGICAL              OMITFK
       COMMON /CNTRL / K5,BDY(6),INSERT,CARRY,CHN
        EQUIVALENCE    (BDY,IBDY)
       COMMON /CPI   / PI,TWOPI,PIO2,PIO4,TODEG,TORAD
       COMMON /CREDIN/ ZTRANS,RTRANS,ROTATE,ZPIVOT,RPIVOT,SCALE,NB,TAB(9)
```

```
      EQUIVALENCE     (XTRANS,ZTRANS),(YTRANS,RTRANS),(XPIVOT,ZPIVOT),
     1                (YPIVOT,RPIVOT)
      COMMON /CTAPOS/ RESTRT,ENDBDT,STCFIL,K6SV
       LOGICAL        RESTRT,ENDBDT,STCFIL
      COMMON /ERASE / B(800)
      COMMON /SPACER/ MAXLH,MAXLT,MAXLF,MAXLW
      COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
       LOGICAL        ERR,ERRMAJ,INERR,PRERR

C     SMOOTH COMMONS
      COMMON /ADAM01/ NAME(6),ADDRES(6),TITLE(6),IDENT(6)
      COMMON /CALCPT/ DX,XMOD
      COMMON /CELLPT / DZETA
      COMMON /CSEGME/ IA(10),IB(10),IMA(10),IMB(10),JTYPE(10),N,NSEG,
     1                NII,NIM
       EQUIVALENCE (NI,NII)
      COMMON /CSMOOA/ DEVA(20),FENDA(20),ANGA(20),CURVA(20),NARB
      COMMON /CSMOOB/ XA(100),YA(100),DEVI(100)
       DIMENSION      ZA(100),RA(100)
       EQUIVALENCE    (ZA,XA),(RA,YA)
      COMMON /CDS2  / X(100),Y(100),ANG(100),ANGD(100),CURV(100),S(100),
     1                FQK(100),DEV(100),CURVB(100)
       DIMENSION      Z(100),R(100),DUM(100)
       EQUIVALENCE    (Z,X),(R,Y),(DUM,CURVB)

      LOGICAL DATAIN,ENDBDC,UPPER,ZRONLY

      DATA KBDY/3HBDY/, KHIGH/6H      /

      NAMELIST /A/         B,        NB,       TAB,     DBLPTS,     ZRONLY,
     1        BDY,       CHN,     UPPER,       X,Z,       Y,R,       ANGD,
     2     ROTATE,    ZPIVOT,    RPIVOT,    ZTRANS,    RTRANS,      SCALE,
     3       FLIP,    XPIVOT,    YPIVOT,    XTRANS,    YTRANS,        DUM
     4,     IDENT,        DX,      XMOD,      DEVA,     FENDA,       ANGA,
     5      CURVA,     ZA,XA,     RA,YA,      DEVI,       NII,        DEV,
     6        ANG,      CURV,     CURVB,       FQK,         S,        NIM,
     7     UPPER

C     DEFINTE DOUBLE POINT TOLERANCE, DPTOL
       DPTOL = 1.E-5

C     INITIALIZE
C      ENDBDC= END OF BDY CARD INPUT, T OR F
       ENDBDC= .FALSE.
       IF(K5.NE.KBDY .OR. ENDCRD) ENDBDC=.TRUE.
   15  DATAIN= .FALSE.
       DBLPTS= .01
       JFOUND= 0
C     READ BDY INPUT CARDS
   35  IF( ENDBDC ) GO TO 40
       FLIP  = 1.
       ROTATE= 0.
       ZPIVOT= 0.
       RPIVOT= 0.
       SCALE = SCALEA
       ZTRANS= 0.
       RTRANS= 0.
       ZRONLY= .FALSE.
```

```
      CALL SETM(1,.1, DEVI,100)
      CALL SETM(3,BITS,XA,200,DEVA,80,B,300)
      CALL SETM(1,BITS,X,200)
      READ  (5,A)
      IF(ZRONLY) CALL ISORT(XA,YA,DUM,B,200,2)
      IF( .NOT.ZRONLY ) CALL ISORT(X,Y,ANGD,B,300,1)
      IF(.NOT.ZRONLY) CALL ISORT(X,Y,ANGA, B,300)
      IF(INERR) ERRMAJ=.TRUE.
      DATAIN= .TRUE.
      RESTRT= .FALSE.

C     COUNT THE LENGTH OF THE  Z-LIST
   40 IF(.NOT.DATAIN) GO TO 900
      IF( JFOUND .EQ.1 ) GO TO 43
      NI    = 0
      DO 41 I=1,100
      IF(XA(I).EQ.BITS) GO TO 42
   41 NI    = I
   42 IF(NI .EQ. 0) GO TO 43
      LINES = 64
      CALL SMOTH
      JFOUND= 1
   43 NZ    = 0
      DO 45 I=1,100
      IF(Z(I).EQ.BITS) GO TO 50
   45 NZ    = I
   50 IF(NZ-2) 55,100,100
   55 WRITE (6,1055) BDY(1)
      ERRMAJ= .TRUE.
      RETURN

C     DELETE DOUBLE POINTS FROM SMOOTH BOUNDARY RECORDS
  100 OMITFK= .TRUE.
      CALL FHEAD(NZ+10)
      WRITE (6,1090) IBDY,CHN,UPPER
      IF(JFOUND.NE.1 .OR. DBLPTS.EQ.0. .OR. NZ.LE.2) GO TO 150
      WRITE (6,1100) DBLPTS,DBLPTS
      I     = 1
  110 I     = I+1
      IF(I.GT.NZ) GO TO 150
      IF(ABS(Z(I)-Z(I-1)).GE.DPTOL .OR.
     1   ABS(R(I)-R(I-1)).GE.DPTOL) GO TO 110
      ANGDIF= ABS(ANGD(I)-ANGD(I-1))
      IF (ANGDIF.GE.DBLPTS) GO TO 110
      NMOVE = NZ-I
      ANGSV = .5*(ANGD(I)+ANGD(I-1))
      IF(ANGD(I)*ANGD(I-1).EQ.0. .AND. ANGDIF.LE..0005) ANGSV=0.
      ANGD(I-1)=ANGSV
      CALL MOVE(3, Z(I+1),Z(I),NMOVE,1,
     1             R(I+1),R(I),NMOVE,1,
     2             ANGD(I+1),ANGD(I),NMOVE,1)
      NZ    = NZ-1
      GO TO 110

C     CALCULATE CURVATURES FOR PRINTOUT
  150 I     = 1
      CURV(1)=0.0
  155 CURVB(I)=BITS
```

```
      CURV(I+1)=CURV(I)
      DX     = Z(I+1)-Z(I)
      DY     = R(I+1)-R(I)
      CHD    = SQRT(DX*DX+DY*DY)
      IF(CHD.LT..00001) GO TO 160
      ACHD   = ATAN3(DY,DX,ANGD(I)*TORAD)
      YPA    = ANG(I)*TORAD-ACHD
      YPB    = ANGD(I+1)*TORAD-ACHD
      CURVB(I)=(4.*YPA+2.*YPB)/(CHD*(1.+1.5*YPA*YPA))
      CURV(I+1)=(-2.*YPA-4.*YPB)/(CHD*(1.+1.5*YPB*YPB))
      GO TO 165
  160 IF(I.EQ.1) GO TO 165
      IF(CURVB(I-1).EQ.BITS) CURVB(I-1)=CURVB(I)
  165 I      = I+1
      IF(I.LT.NZ) GO TO 155
      CURVB(I)=0.0
*RELO13     RELOCATE FROM A ONE TO A THREE DIMENSIONED ARRAY -RELO13-
C     SUBROUTINE RELO13

C     INPUT-
C      Z,R    = BOUNDARY COORDINATES
C      ANGD   = ANGLE OF THE BOUNDARY (DEGREES)
C      NZ     = NUMBER OF BOUNDARY COORDINATE POINTS
C      FLIP   = SCALER ON R(I) BEFORE ROTATION OR TRANSLATION
C      ROTATE= ANGULAR ROTATION IN DEGREES
C      ZPIVOT,RPIVOT=PIVOT POINT FOR ROTATION BEFORE SCALING
C      SCALE  = MULTIPLICATIVE CONSTANT ON INPUT COORDINATES
C      ZTRANS= Z-TRANSLATION AFTER SCALING
C      RTRANS= R-TRANSLATION AFTER SCALING
C      BDY    = BOUNDARY NAME
C      UPPER  = T IF UPPER BOUNDARY, = F IF LOWER BOUNDARY
C      CHN    = CHANNEL NAME
C      LBDE   = NEXT AVAILABLE LOCATION IN THE BOUNDARY TABLE

C     OUTPUT-
C      BDT    = TABLE OF Z,R,ANG  IN 3-D ARRAY FORM
C      LBDE   = NEXT AVAILABLE LOCATION IN THE BOUNDARY TABLE

      IF(FLIP.NE.1..OR. ROTATE.NE.0. .OR. SCALE.NE.1. .OR. ZTRANS.NE.0.
     1 .OR. RTRANS.NE.0.) WRITE (6,1151) FLIP,ROTATE,ZPIVOT,RPIVOT,
     2 SCALE,ZTRANS,RTRANS
      WRITE (6,1152)
      LB1    = LBDE
      LB2    = LB1+3*(NZ-1)
      LB     = LB1
      BDT(LB)=BDY
      CHNAME(LB)=CHN
      LBZ1(LB)=0
      UP(LB)= UPPER
      LEDEX(LB)=0
      I      = 1
      LBDEL  = 3
      ADDPI  = 0.
      IF(.NOT.UPPER) GO TO 240
      LB     = LB2
      LBDEL  = -3
      ADDPI  = PI
  240 ROTAT  = ROTATE*TORAD
```

78

```
      SN     = SIN(ROTAT)
      CS     = COS(ROTAT)
  250 IF(ROTATE.NE.0.) GO TO 260
      ZBT(LB)=Z(I)*SCALE + ZTRANS
      RBT(LB)=R(I)*FLIP*SCALE + RTRANS
      GO TO 270
  260 RFLP   = R(I)*FLIP
      ZBT(LB)=(ZPIVOT+CS*(Z(I)-ZPIVOT)-SN*(RFLP-RPIVOT))*SCALE + ZTRANS
      RBT(LB)=(RPIVOT+CS*(RFLP-RPIVOT)+SN*(Z(I)-ZPIVOT))*SCALE + RTRANS
  270 ANGD(I)=ANGD(I)*FLIP + ROTATE
      ANGBT(LB)=ANGD(I)*TORAD + ADDPI
      WRITE (6,1280) I,ZBT(LB),RBT(LB),ANGD(I),CURV(I),CURVB(I)
      IF(I.GE.NZ) GO TO 300
      I      = I+1
      LB     = LB+LBDEL
      GO TO 250
  300 LBDE   = LB2+9
      LBNEXT(LB1)=LBDE-LB1
      BDT(LBDE)=BLANK
C     END SUBROUTINE RELO13


  900 RETURN

 1055 FORMAT(//1X43H**   NO COORDINATE INPUT WAS FOUND FOR BDY=A6,//)
 1090 FORMAT(///1X,45HB O U N D A R Y  C O O R D I N A T E S,  BDY=A6,
     * 5X4HCHN=A6,5X6HUPPER=
     *L2,)
 1100 FORMAT(/6X46HDOUBLE POINTS WITH ANGLE DIFFERENCES LESS THANF6.3,1X
     *24HARE ELIMINATED  (DBLPTS=F5.3,2H).)
 1151 FORMAT(/6X5HFLIP=F7.3,3X7HROTATE=F8.3,3X7HZPIVOT=F10.5,3X7HRPIVOT=
     *F11.5,3X5HSCALEF7.3,3X7HZTRANS=F10.5,3X7HRTRANS=F10.5,)
 1152 FORMAT/9X48HI    X,Z        Y,R        ANGD      CURV-      CURV+)
 1280 FORMAT(I10,2F10.5,F10.3,2F10.4)
      END
```

```
*DECK CRBD
      BLOCK DATA RBDBLK
*CRBD--          BLOCK DATA FOR RBD ROUTINE                          -CRBD-
*SMOBLK          SMOOTH BLOCK COMMON                                 -SMOBLK-
      COMMON /CSMOOD/ SGAMMA,SZETA1,SZETAN
      DATA SGAMMA,SZETA1,SZETAN/  1.,1.E2,1.E2/
      END
```

```
*DECK RCD
      SUBROUTINE RCD
*RCD---          READ IN CHANNEL DATA                                -RCD-

C     INPUT-
C      CHDATA- CHANNEL INPUT DATA TABLE
C      LHE    = NEXT AVAILABL LOCATION IN CHANNEL INPUT DATA TABLE
C
C     OUTPUT-
C      LCHE   = NEXT AVAILABL LOCATION IN CHANNEL INPUT DATA TABLE
C      CHDATA- CHANNEL INPUT DATA TABLE INCLUDING NEW INPUT VALUES

       COMMON           PROGM(8),PROGSV,FILIN,FILOT,REFS(5)
        LOGICAL                          FILIN,FILOT
       COMMON /CBITS / BITS,BLANK
       COMMON /CNTRL / K5,CHN(6),INSERT
        INTEGER CHN
        EQUIVALENCE    (ICHN,CHN)
       COMMON /CTABPR/ IITAB
       COMMON /CTAPOS/ RESTRT,ENDBDT,ENDFIL,K6SV
        LOGICAL        RESTRT,ENDBDT,ENDFIL
       COMMON /SPACER/ MAXLH,MAXLT,MAXLF,MAXLW
       COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
        LOGICAL        LRR,ERRMAJ,INERR,PRERR

C     CHANNEL INPUT DATA TABLE
C      INDEX- LH=LHO,LHE
       COMMON /CHDATA/ CHNAM(1),LHNEXT(1),WTFLOW(1),TTO(1),PTO(1),
      1                TSO(1),PSO(1),MACHO(1),AO(1),VARY(1),
      2                RG(1),GAM(1), NR(1),NC(1),TAB(6),
      4                BB(75)
        LOGICAL        VARY
        INTEGER CHNAM
        DIMENSION      VO(1)
        REAL           MACHO
        EQUIVALENCE    (VO,MACHO)
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
        DIMENSION      LIMITS(24)
        EQUIVALENCE    (LIMITS,LHO)

       COMMON /ERASE / DUM(16),B(784)

       NAMELIST /A/ CHN,WTFLOW,TTO,TI,PTO,PT,
      1             TSO,PSO,MACHO,AO,VARY,
      2             GAM,RG,
      3             NR,NB,TAB,B

C     RESTART CASE WITH CHANNEL FLOW DATA REVISIONS
C      RELOCATE CHDATA FOR CHANNEL=CHN INTO FIRST POSITION
C      FIRST FIND INDEX  LH  FOR CHNAM=CHN
       LH     = LHO
   12  IF(LH.GE.LHE) GO TO 20
       IF(CHNAM(LH).EQ.CHN) GO TO 14
       LH     = LH+LHNEXT(LH)
```

75

```
      GO TO 12
   14 IF(LH.EQ.LHO) GO TO 16
      LNG    = LHNEXT(LH)
      LH1    = LHO+LNG
      LH2    = LH+LNG
      LH3    = LH2+LNG
      CALL MOVE(3, CHNAM(LHO),CHNAM(LH1),LHO-LHE-1,1,
     1             CHNAM(LH2),CHNAM(LHO),LNG,1,
     2             CHNAM(LH3),CHNAM(LH2),LHE+LNG-LH3+1,1)
   16 LHNXT = LHO+LHNEXT(LHO)
      GO TO 30

   20 CALL MOVE(1, CHNAM,CHNAM(21),LHO-LHE-1,1)
      LHNEXT= 20
      LHNXT = 21
      LHE    = LHE+20

C     INITIALIZE
      CALL SETM(1,BITS,WTFLOW,10)
      VARY   = .TRUE.

C     READ  CHN  INPUT CARDS
   30 CALL SETM(1,BITS, B,400)
      READ (5,A)
      IF(INERR) ERRMAJ=.TRUE.

C     RESET  CHNAM  IF CHANNEL NAME HAS BEEN REDEFINED
      CHNAM = CHN

C     COUNT THE LENGTH OF THE B-ARRAY
      NR     = 0
      NC1    = NC
      DO 40 I=1,400,NC1
      IF(B(I).EQ.BITS) GO TO 50
   40 NR     = NR+1
   50 NCR    = NC*NR

C     RELOCATE AND INSERT B-ARRAY INTO CHDATA-TABLE
      IF(NCR.EQ.0) GO TO 950
      LHNXTT= LHO+20+NCR
      NMOVE = LHE-LHNXT+1
      IF(LHNXTT.GT.LHNXT) NMOVE=-NMOVE
      CALL MOVE(2, CHNAM(LHNXT),CHNAM(LHNXTT),NMOVE,1, B,BB,NCR,1)
      LHE    = LHE+LHNXTT-LHNXT
      LHNEXT= 20+NCR

  950 IF(LHE.LT.LBDO) GO TO 980
      WRITE (6,1960) LHO,LHE,MAXLH,LBDO
      CALL ERROR1
  980 RETURN
 1960 FORMAT(/1X81H***  THE CHANNEL INPUT DATA TABLE HAS EXCEEDED ALLOTT
     *ED MEMORY.  INCREASE  MAXLH./6X4HLHO=I4,3X4HLHE=I4,3X6HMAXLH=I4,3X
     *5HLBDO=I4,)
      END
```

```
*DECK REDINP
      SUBROUTINE REDINP
*REDINP         STC READ INPUT                                        -REDINP-

       COMMON /BCOMMN/ PROGM(8),PROGSV,FILIN,FILOT
        LOGICAL                        FILIN,FILOT
       COMMON /ADAM01/ NAME(6),ADDRES(6),TITLE(6),IDENT(6)
       COMMON /ADAM02/ ENDJOB,NUMPLT,PLOTED,ENDCRD
        LOGICAL        ENDJOB,       PLOTED,ENDCRD
       COMMON /ALLCOM/ MACHA,PSA,TSA,PTA,TTA, AXIA,RGA,GAMA,
      1                MACHC,PSC,TSC,PTC,TTC, AXIC,RGC,GAMC,
      2                DAXIT,SCALEA,TTE,CHOTST
        REAL           MACHA(1),MACHC
        LOGICAL        AXIA,AXIC
        LOGICAL        CHOTST
        LOGICAL        AXI
        REAL           MACHO(1)
        EQUIVALENCE    (MACHO,MACHC),(PSO,PSC),(TSO,TSC),(PTO,PTC),
      1                (TTO,TTC), (AXI,AXIC),(RG,RGC),(GAM,GAMC)
       COMMON /CSS    / SSFML,SSEF,SSEANG,SSDF,SSFEND,SSFND1
      1                ,SSDLE,A4FACT,BRLX,CURRLX,TSIC,RHOC,RHOCSS
        INTEGER        SSFML
        LOGICAL                SSEF,       SSDF,   SSDLE
C       SSFML = SUPERSONIC CURVATURE FORMULA NUMBER
C       SSEF  = SUPERSONIC ENTERING FLOW, T OR F
C       SSEANG= ENTERING FLOW ANGLE (DEGREES) FOR SSEF=T
C       SSDF  = SUPERSONIC DISCHARGE FLOW, T OR F
C       SSFEND= SUPERSONIC BEAM DOWNSTREAM END CONDITION, =0,1 FOR PARABOL
C       SSFND1= SUPERSONIC BEAM UPSTREAM END CONDITION, =0,1, FOR PARABOLA
C       SSDLE = SS FLOW BELOW AND AFT OF LE PT, T OR F
C       A4FACT= CENTRAL POINT INFLUENCE COEFFICIENT FACTOR
C       BRLX  = B-RELAXATION FACTOR
C       CURRLX= CURVATURE RELAXATION FACTOR
C       RHOC  = ACCELERATION FACTOR ON CURVATURE ITERATION AT
C               SUBSONIC POINTS
C       RHOCSS= ACCELERATION FACTOR ON CURVATURE AT SUPERSONIC POINTS
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
        DIMENSION      LIMITS(24)
        EQUIVALENCE    (LIMITS,LHO)
       COMMON /SLTAB / W(128),X2(128),SLCHN(128)
        INTEGER SLCHN
       COMMON /BENDIN/ NBCIN(2),ACF(2)
       COMMON /CB     / B(300)
       COMMON /CBITS / BITS,BLANK
       COMMON /CCRX   / CRXSL,CRXOL,CRXSS,CRXE,CRXC,CRMACH
        DIMENSION      CRX(6)
        EQUIVALENCE    (CRX,CRXSL)
       COMMON /CHDATA/ TABLES(2046)
       COMMON /CEND   / TBLEND(2)
       COMMON /CIADIN/ RHOBAS,RHOAMP,IADM
       COMMON /CINNER/ INRCTR,RDUM,NINNER(16),CNVF(16)
       COMMON /CISBOT/ FARFLD(2),FREE(2),PRES(2),RFF,NZP,
      1                ZP(10),PPS(10), A1,A2,ADUM(6)
        INTEGER        FARFLD,FREE,PRES
       COMMON /CLINES/ LINES,OMITFK,PTITLE(6)
```

17

```
      COMMON /CLWOSV/ LWOSV
      COMMON /CM     / JMS(300)
      COMMON /CMAXIT/ MAXIT,MAJCTR,GREFIN,EDUM
       LOGICAL                     GREFIN
      COMMON / CNORM / RHL,RM,AHL,ARM
      COMMON /CNTRL / K5(1),STA(6),INSERT
      COMMON /CPRINT/ PRTES2,PRTB,PRTA,PREFIN,PREFN2,SSONIC,PDUM(20)
      COMMON /CPRPRN/ PRPRN
       INTEGER        PRPRN
      COMMON /CPTMOV/ VELPOT,ICOB,NODENS,FBASTG
       LOGICAL        VELPOT
      COMMON /CR     / RF(300)
      COMMON /CREFIN/ SLS,SG21,VMG1,VMG2, NGR,NGZ,SGR(10),GR(10),
     1                SGZ(10),GZ(10)
      COMMON /CS1    / S1(300)
      COMMON /CS2    / S2(300)
      COMMON /CTAPOS/ RESTRT,ENDBDT,STCFIL,K6SV
       LOGICAL        RESTRT,ENDBDT,STCFIL
      COMMON /CTHICK/ NTHKX,NTHKY,THKX(20),THKY(20),THIK2D(78)
      COMMON /CTOLRL/ TOLRL,MAXSWP,CLEN,DS2MX,TOLES2,NSWP,
     1                DS1DMP,DS1MXA,DS1MXB,DS1RMS,ES2MX
     *               ,DS1RM0,SG1MIN,TOLINR
      COMMON /CVM    / VMF(300)
      COMMON /CZ     / ZF(300)
      COMMON /SPACER/ MAXLH,MAXLT,MAXLF,MAXLW
      COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
       LOGICAL        ERR,ERRMAJ,INERR,PRERR
      COMMON /TAPES / NTAPO,NTAPN


      LOGICAL         FIRST

      DATA KA/1HA/, KBDY/3HBDY/, KCHN/3HCHN/, KSTA/3HSTA/
      DATA FIRST/.TRUE./

C     ENDCRD= T IF END OF CARD INPUT
C     ENDBDT= T IF END OF BOUNDARY DATA ON TAPE
C     STCFIL= T IF A STC-SUBFILE EXISTS ON TAPE=ORGF.

      NAMELIST /A/ IDENT, AXI, RG,GAM, MACHO,PSO,TSO, PTO,TTO, PRPRN,
     1       INRCTR,TTE,CHOTST,MAXIT,MAJCTR,NINNER,VELPOT,ICOB,NODENS,RN,
     2              VMG1,VMG2, NGR,NGZ, SGR,GR, SGZ,GZ, SLS,SG21,
     3              NBCIN,ACF, SSFML,SSEF,SSEANG,SSDF,SSFEND,SSFND1,
     4              SSDLE, A4FACT, BRLX, CURRLX, TSIC, RHOC, RHOCSS,
     5              FARFLD,FREE,PRES,RFF,NZP,ZP,PPS,A1,A2,ADUM,
     6              LIMITS,TABLES, B,JMS,S1,S2,ZF,RF,VMF, W,X2,SLCHN,
     8              TOLRL,MAXSWP,TOLES2,TOLINR,SG1MIN,DS1DMP,DS1RM0,
     9              CRXSL,CRXOL,CRXSS,CRXE,CRXC,CRX,
     *                 PRTES2,PRTB,PRTA,PREFIN,PREFN2,SSONIC,PDUM,
     *              MAXLH,MAXLT,MAXLF,MAXLW, KEYB,  RDUM, CNVF,
     *              PLOT,IPLOT,SAMEXY,XSCALE,YSCALE,
     A              RHOBAS,RHOAMP,IADM,
     B              NTHKX,NTHKY,THKX,THKY,THIK2D,
     C              LBL,MAXLBL,TOLLBL

C**   INITIALIZE AND READ OVERALL (A) INPUT DATA
      IF(.NOT.FIRST .AND. (K5.NE.KA .OR. ENDCRD)) GO TO 200
      IF(FIRST .AND. K5.EQ.KA) GO TO 100
```

```
      WRITE (6,1000)
      ERR    = .TRUE.
      PROGSV= 0.
      GO TO 200

  100 PROGSV= 0.
      ENDBDT= .FALSE.
      FIRST = .FALSE.
      LINES = 64
      MAJCTR= 0
      RESTRT= .TRUE.
      STCFIL= .FALSE.
      CALL SETM(1,BITS, MACHO,8)

C     DETERMINE FIELD ARRAY SIZE
      MAXLE = LOC2(TABLES,TBLEND)
      MAXNM = LOC2(RF,ZF)
      GO TO 120

C    READ CARD INPUT
  130 READ (5,A)
      DO 135 I=1,8
  135 IF(MACHO(I).NE.BITS) MACHA(I)=MACHO(I)
      DATA A00000/6HA00000/
      K6SV   = A00000

C    DEFINE THE CHARACTERISTIC LENGTH, CLEN
      CLEN   = SGR
      IF(NGR.LE.1) GO TO 146
      DO 144 I=2,NGR
  144 CLEN   = CLEN+SGR(I)
  146 IF(NGZ.LE.0) GO TO 149
      DO 148 I=1,NGZ
  148 CLEN   = CLEN+SGZ(I)
  149 CLEN   = CLEN/FLOAT(NGR+NGZ)

C    SET UP INDEX-ORIGIN TABLE IF THERE IS NO  STC-TAPE INPUT
C     ORDER OF TABLES IN BLOCK COMMON
C      LH     /CHDATA/
C      LB     /BDYTAB/
C      LT     /CONVTB/
C      LW     /WAKETB/
C      LF     /CADJWF/
C      L      /STATAB/
      IF(STCFIL) RETURN
      RESTRT= .FALSE.
      LBDO   = LHO+MAXLH
      LBDE   = LBDO
C     (OTHER INDEX LIMITS ARE SET IN SUBROUTINE BLDTBS)
      RETURN

C    READ INPUT FILE
  120 IF(.NOT.FILIN) GO TO 130
      REWIND NTAPO
      READ (NTAPO)  STCFIL,(LIMITS(I),I=1,24)
       LWOSV = LWO
      IF(STCFIL) GO TO 125
      ENDBDT= .TRUE.
```

```
      WRITE (6,1120)
      GO TO 130

  125 READ (NTAPO) ((IDENT(I),I=1,6),AXI,RG,GAM,MACHO,PSO,TSO,PTO,TTO,
     1  PRPRN,TTE,CHOTST,MAXIT,MAJCTR, (NINNER(I),I=1,16), VELPOT,ICOB,
     2  NODENS,RN,NGR,NGZ,(SGR(I),I=1,40),VMG1,VMG2, INRCTR, SLS,SG21,
     3  NBCIN(1),NBCIN(2),ACF(1),ACF(2), SSFML,SSEF,SSEANG,SSDF,SSFEND,
     4  SSFND1,SSDLE,A4FACT,BRLX,CURRLX,TSIC,(FARFLD(I),I=1,8),
     * RHOC,RHOCSS,RHL,RM,
     5  (ZP(I),I=1,28), (TABLES(I),I=1,LESTA), (B(I),I=1,NM), (JMS(I),
     6  I=1,NM), (S1(I),I=1,NM), (S2(I),I=1,NM),(ZF(I),I=1,NM), (RF(I),
     7  I=1,NM), (VMF(I),I=1,NM), (W(I),I=1,NJ), (X2(I),I=1,NJ),
     8  (SLCHN(I),I=1,NJ),TOLRL,MAXSWP,TOLES2,TOLINR,SG1MIN,DS1DMP,
     A  DS1RMO,(CRX(I),I=1,6), RHOBAS,RHOAMP,IADM,NTHKX,NTHKY,
     B  (THKX(I),I=1,118)  )
C       CHECK TO SEE IF STC-A INPUT DATA EXCEEDED DIMENSIONS
      IF(NM.GT.LOC2(RF,ZF) .OR. LESTA.GT.LOC2(TABLES,TBLEND)) ERR=.TRUE.
      GO TO 130

C     READ BOUNDARY DATA
  200 CALL RBD
      IF(ENDCRD) GO TO 700
      IF(K5.EQ.KBDY) RETURN

C     READ CHANNEL DATA
  300 IF(K5.NE.KCHN) GO TO 400
C       IF RESTRT, UNPACK TABLES TO MAKE ROOM FOR NEW CHDATA AND CONVTB.
        IF(.NOT.RESTRT .OR. LBDO.GT.(LHE+1)) GO TO 350
        MOVE1 = LOC2(TABLES,S1)-LESTA
        MOVE2 = MOVE1/2
        LWTO  = LWO+MOVE1
        LBTO  = LBDO+MOVE2
        CALL MOVE(2, TABLES(LWO),TABLES(LWTO),LWO-LESTA-1,1,
     1                TABLES(LBDO),TABLES(LBTO),LBDO-LTE-1,1)
        LBDO  = LBDO+MOVE2
        LTE   = LTE+MOVE2
        LBDE  = LBDE+MOVE2
        LTO   = LTO+MOVE2
        LWO   = LWO+MOVE1
  350 CALL RCD
      RETURN

  400 WRITE (6,1690)
      ERRMAJ= .TRUE.
      RETURN

C     CONSTRUCT LETEPT, ORTCHN, CONVTB, SLTAB, STATAB AND THE FIELD TABLE
  700 IF(ERRMAJ .OR. LBDE.EQ.LBDO) ERR=.TRUE.
  900 RETURN
 1000 FORMAT(/1X73HERROR- THE K5=A INPUT DATA DOES NOT IMMEDIATELY FOLLO
     *W THE PROGM=STC CARD)
 1120 FORMAT(//1X43H***  NO  STC  DATA FOUND ON THE INPUT TAPE.//)
 1690 FORMAT(//1X44H**   PLEASE CHECK THE INPUT VALUE OF K5 (K5=A6,18H).
     *  IT MUST BE ONE/6X37HOF THE FOLLOWING-   A, BDY, CHN, STA.//)
      END
```

```
*DECK RELOXY
      SUBROUTINE RELOXY(I1,I2, NPTS, IM1,IM2)
*RELOXY          RELOCATE X,Y,ANG,ANGD,CURV,S,FQK                -RELOXY-

C     INPUT-
C      I1,I2 = INDEX RANGE OF SEGMENT DATA IN XA,YA-ARRAYS
C      NPTS  = NO OF PTS REQD FOR SEGMENT DEFINITION IN X,Y-ARRAYS
C      IM1   = INDEX OF FIRST POINT OF THE SEGMENT IN X,Y-ARRAYS
C      IM2   = INDEX OF LAST  POINT OF THE SEGMENT IN X,Y-ARRAYS
C      NIM   = LENGTH OF X,Y-ARRAYS
C      N     = SEGMENT INDEX

C     OUTPUT-
C      IM2   = INDEX OF LAST POINT IN RELOCATED X,Y-ARRAYS
C      RELOCATED X,Y,..-ARRAYS
C      ADJUSTED IMA,IMB INDEX LIMIT VALUES

      COMMON /CSEGME/ IA(10),IB(10),IMA(10),IMB(10),JTYPE(10),
     1                N,NSEG, NI,NIM
      COMMON /CDS2   / X(100),Y(100),ANG(100),ANGD(100),CURV(100),
     1                S(100),FQK(100),DEV(100),CURVB(100)
      COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
       LOGICAL        ERR,ERRMAJ,INERR,PRERR

      NADD  = NPTS - (I2-I1+1)
      IF    = IM2+1
      IT    = IF+NADD
      NMOVE = NIM-IM2
      IF(NADD.GE.0) NMOVE=-NMOVE
      NIM   = NIM+NADD
      IF(NIM.LE.100) GO TO 30
      ERR   = .TRUE.
      WRITE (6,1030)
      RETURN
 1030 FORMAT(/1X67HSORRY - THE NO. OF OUTPUT PTS. EXCEEDS THE ALLOCATED
     *STORAGE (200).)
   30 IF(NMOVE*NADD.EQ.0) GO TO 50
      CALL MOVE(3, X(IF),X(IT),NMOVE,1,
     1             Y(IF),Y(IT),NMOVE,1,
     2             ANG(IF),ANG(IT),NMOVE,1)
      CALL MOVE(3, ANGD(IF),ANGD(IT),NMOVE,1,
     4             CURV(IF),CURV(IT),NMOVE,1,
     5             S(IF),S(IT),NMOVE,1)
      CALL MOVE(3, FQK(IF),FQK(IT),NMOVE,1,
     7             DEV(IF),DEV(IT),NMOVE,1,
     8             CURVB(IF),CURVB(IT),NMOVE,1)
   50 IM2   = IM1 + NPTS-1
      IF(IM2.LT.IM1) GO TO 70
      DO 60 I=IM1,IM2
      DEV(I)= 0.
      CURVB(I)=0.
   60 FQK(I)= 0.
   70 IMB(N)= IM2
      NP1   = N+1
      IF(NP1.GT.NSEG) GO TO 900
      DO 80 NN=NP1,NSEG
      IMA(NN)=IMA(NN)+NADD
   80 IMB(NN)=IMB(NN)+NADD
```

```
*DECK SERS1
      SUBROUTINE SERS1(X1,Y1, X2,Y2, A)
*SERS1-         NACA SERIES-1 COWL CONTOUR                    -SERS1-

C     INPUT-
C      X1,Y1 = COORDINATES AT HIGHLITE
C      X2,Y2 = COORDINATES ON COWL SURFACE
C      A     = X/X LIMIT POINT

C     OUTPUT-
C      CALC VALUES OF X,Y,ANG,ANGD,CURV,S

       COMMON /CBEND / NBC(2),ANGE(2),CURVE(2),FEND(2)
       COMMON /CPI   / PI,TWOPI,PIQ2,PIQ4,TODEG,TORAD
       COMMON /CSEGME/ IA(10),IB(10),IMA(10),IMB(10),JTYPE(10),
      1                N,NSEG, NI,NIM
       COMMON /CDS2  / X(100),Y(100),ANG(100),ANGD(100),CURV(100),
      1                S(100),FOK(100),DEV(100),CURVB(100)
        DIMENSION      ANGB(100)
        EQUIVALENCE    (ANGB,CURVB)

       DIMENSION       XS1(40),YS1(40),TS1(40)

       DATA XS1/
      *0.,.000106,.0003062,.0006461,.0012998,.0020031,.0039664,.006002,
      *.008,.01,.015,.02,.025,.03,.035,.04,
      *.045,.05,.06,.07,.08,.09,.1,.12,
      *.14,.16,.18,.20,.22,.25,.3,.35,
      *.4,.45,.5,.6,.7,.8,.9,1.0/
       DATA YS1/
      *0.,.0112,.019,.0275,.0388,.047969,.066707,.08117,
      *.093118,.10386,.127271,.147458,.165786,.182977,.199304,.214829,
      *.229594,.243677,.270135,.29478,.318041,.340196,.361381,.40087,
      *.43654,.468883,.498788,.526959,.553714,.591484,.648994,.700757,
      *.74746,.789479,.827209,.89087,.939554,.973716,.993649,1./
       DATA TS1/
      *0.,52.52592,30.79679,21.04343,14.69820,
      *11.71671,7.996274,6.397164,5.618328,5.133687,
      *4.308968,3.821510,3.533277,3.342515,3.183152,
      *3.029897,2.884790,2.755270,2.545330,2.388930,
      *2.268497,2.165982,2.068093,1.875127,1.697514,
      *1.552614,1.446208,1.368108,1.303797,1.217213,
      *1.090491,.981545,.885102,.797345,.715438,
      *.560407,.412448,.269017,.13063,0./

C     DETERMINE CUT-OFF POINT, NPTS
       IF(.05.LE.A .AND. A.LE.1.) GO TO 50
       WRITE (6,1050) A
       CALL ERROR1
   50  DO 60 K=17,40
       IF(XS1(K).GT.A) GO TO 70
   60  NPTS  = K

C     RELOCATE ARRAYS
   70  I1    = IA(N)
       I2    = IB(N)
       IM1   = IMA(N)
       IM2   = IMB(N)
```

83

```
      CALL RELOXY(I1,I2, NPTS, IM1,IM2)
      XR     = X2-X1
      YR     = Y2-Y1
      AR     = YR/XR

      K      = 1
      DO 120 I=IM1,IM2
      X(I)   = X1+XR*XS1(K)
      Y(I)   = Y1+YR*YS1(K)
      IF(I.EQ.IM1) GO TO 115
      ANG(I)= ATAN(AR*TS1(K))
      GO TO 118
  115 ANG(I)=PIQ2
  118 ANGD(I)=ANG(I)*TODEG
  120 K      = K+1

      NBC(1)= 1
      NBC(2)= 1
      ANGE(1)=ANGD(IM1)
      ANGE(2)=ANGD(IM2)
      ANGB(IM1)=ANG(IM1)
      CALL BFACES(X,Y,ANGB,CURV,FQK,S, IM1,IM2)

      CALL FHEAD(51)
      WRITE (6,1150) X1,Y1,X2,Y2,A
      K      = 1
      DO 160 I=IM1,IM2
      ANGB(I)=ANGB(I)*TODEG
      WRITE (6,1160)
     * XS1(K),YS1(K),X(I),Y(I),ANGD(I),ANGB(I),CURV(I),S(I)
  160 K      = K+1
      CALL MOVE(1,CURV(IM1),CURVB(IM1),K-1,1)
      RETURN

 1050 FORMAT(/1X70H*** INPUT ERROR, PARAMETER A DOES NOT SATISFY  .05-A-
     *1.0 CRITERIA,  A=F6.3,)
 1150 FORMAT(/22X,30H* NACA SERIES-1 COWL CONTOUR *//4X16HINPUT DATA,  X
     *1=F9.5,3X3HY1=F9.5,/17X3HX2=F9.5,3X3HY2=F9.5,3X2HA=F6.3,///4X16HCO
     *ORDINATE DATA-//71X,29H-----  BEAM CALCULATED  -----/10X3HX/X7X,3H
     *Y/Y14X,1HZ14X,1HR9X,35HANGD         ANGB       CURV          S)
 1160 FORMAT(7X,F8.6,F10.5,F16.5,F15.5,F11.3,F12.3,F11.6,F10.5,)
      END
```

```
*DECK SMOINP
      SUBROUTINE SMOINP
*SMOINP         INPUT/OUTPUT AND SPECIAL CONTOUR ROUTINE    -SMOINP-
      COMMON           PROGM(8),PROGSV,FILIN,FILOT,REFS(5)
       LOGICAL                                FILIN,FILOT
      COMMON /ADAMO1/ NAME(6),ADDRES(6),TITLE(6),IDENT(6)
      COMMON /CALCPT/ DX,XMOD
      COMMON /CBITS / BITS,BLANK
      COMMON /CELLPT/ DZETA
      COMMON /CLINES/ LINES,OMITFK,PTITLE(6)
       LOGICAL              OMITFK
      COMMON /CNTRL / K5(1),STA(2),INCLUD(2),DELETE(2),INSERT,CARRY
       LOGICAL                                                CARRY
      EQUIVALENCE     (BDY,STA)
      COMMON /CPI   / PI,TWOPI,PIQ2,PIQ4,TODEG,TORAD
      COMMON /CSEGME/ IA(10),IB(10),IMA(10),IMB(10),JTYPE(10),
     1                N,NSEG, NII,NIM
       EQUIVALENCE    (NI,NII)
      COMMON /CSMOOA/ DEVA(20), FENDA(20),ANGA(20),CURVA(20), NARB
      COMMON /CSMOOB/ XA(100),YA(100),DEVI(100)
       DIMENSION      ZA(100),RA(100)
       EQUIVALENCE    (ZA,XA),(RA,YA)
      COMMON /CDS2  / X(100),Y(100),ANG(100),ANGD(100),CURV(100),
     1                S(100),FQK(100),DEV(100),CURVB(100)
       DIMENSION      DUM(100)
       EQUIVALENCE    (DUM,CURVB)
       DIMENSION      Z(100),R(100)
       EQUIVALENCE    (Z,X),(R,Y)
      COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
       LOGICAL        ERR,ERRMAJ,INERR,PRERR, ERRCAS
       EQUIVALENCE    (ERRCAS,INERR)

      LOGICAL         UPPER

      DIMENSION       CNAMES(4)
      DATA CNAMES/990.,992.,993.,991./

C*** DEFINE THE NUMBER OF SEGMENTS AND THE INDEX LIMITS
C     NSEG  = NUMBER OF SEGMENTS
C     N     = SEGMENT INDEX
C     IA(N),IB(N)=LIMITS OF SEGMENT IN THE XA,YA LISTS
C     TYPE(N)=TYPE OF SEGMENT
   45 N     = 1
      I     = 1
      IJUNCT= 1
      GO TO 55
   50 IF(XA(I).EQ.XA(I-1) .AND. YA(I).EQ.YA(I-1)) GO TO 70
   55 IF(I-NI) 60,155,155
   60 DO 65 J=1,4
   65 IF(XA(I).EQ.CNAMES(J)) GO TO 75
      IF(I.EQ.IJUNCT) GO TO 70
      I     = I+1
      GO TO 50

C    CONTOUR JUNCTURE
   70 J     = 1
   75 JTYPE(N)=J
      IA(N) = I
```

65

```
      N      = N+1
      GO TO (110,120,130,140),J

C     ARBITRARY CURVE
  110 IB(N-1)=0
      I      = I+1
      GO TO 50

C     ELLIPSE
  120 IB(N-1)=I+3
      IF((I+2.EQ.NI) .OR. (XA(I+2).EQ.XA(I+3) .AND. YA(I+2).EQ.YA(I+3)))
     +IB(N-1)=I+2
      GO TO 150

C     SPIRAL
  130 IB(N-1)=I+3
      GO TO 150

C     SERIES 1
  140 IB(N-1)=I+2
  150 I      = IB(N-1)+1
      IJUNCT= I
      GO TO 55

C     END OF INPUT DATA, FILL ZERO IB(N)
  155 NSEG   = N-1
      IB(N-1)=NI
      DO 160 N=1,NSEG
  160 IF(IB(N).EQ.0) IB(N)=IA(N+1)-1
      RETURN

C*** FIT THE SPECIAL CONTOURS
      ENTRY CONTRS
      DO 195 N=1,NSEG
      IMA(N)= IA(N)
  195 IMB(N)= IB(N)
      NIM    = IB(NSEG)
      N      = 1
  200 J      = JTYPE(N)
      IF(J.LE.1) GO TO 790
      OMITFK= .TRUE.
      CALL FHEAD(6)
      WRITE (6,1202) N,BDY
      I      = IA(N)
      I2     = IB(N)
      IM     = IMA(N)
      IM2    = IMB(N)
      X1     = XA(I+1)
      Y1     = YA(I+1)
      IF(N.LE.1) GO TO 206
      X1     = X(IM-1)
      Y1     = Y(IM-1)
  206 X2     = XA(I+2)
      Y2     = YA(I+2)
      IF(N.EQ.NSEG .OR. JTYPE(N+1).NE.1) GO TO 220
      X2     = X(IM2+1)
      Y2     = Y(IM2+1)
  220 IF(IM.LE.1) GO TO 222
```

```
      ANG1   = ANGD(IM-1)
  222 IF((I2-I).EQ.3 .AND. (XA(I+3).NE.BITS.AND.XA(I+3).NE.999.))
     *         ANG1=XA(I+3)
      IF(IM2.GE.NIM) GO TO 224
      ANG2   = ANGD(IM2+1)
  224 IF((I2-I).EQ.3 .AND. (YA(I+3).NE.BITS .AND. YA(I+3).NE.999.))
     *         ANG2=YA(I+3)
      IF(J-3) 250,300,400

C     FIT THE ELLIPSE
  250 CALL ELLIP(X1,Y1,ANG1, X2,Y2,ANG2, YA(I))
      IF(ERR) GO TO 790
      DZETA = 5.*TORAD
      CALL ELLIPT
      GO TO 790

C     FIT THE HYPERBOLIC SPIRAL
  300 IF(YA(I).EQ.2.) GO TO 320
      CALL HYPER1(X1,Y1,ANG1, X2,Y2,ANG2)
      GO TO 350
  320 CURV1 = YA(I+3)
      CALL HYPER2(X1,Y1,ANG1,CURV1, X2,Y2)
  350 IF(ERR) GO TO 790
      CALL HYPTS
      GO TO 790

C     SERIES 1 COWL LIP.
  400 CALL SERS1(X1,Y1, X2,Y2, YA(I))

C     INDEX TO THE NEXT SEGMENT
  790 IF(ERR) ERRCAS=.TRUE.
      ERR    = .FALSE.
      N      = N+1
      IF(N.LE.NSEG) GO TO 200

C     IF ERR HAS BEEN ENCOUNTERED, DO NOT WRITE OUTPUT FILE
      IF(.NOT.ERRCAS) GO TO 800
      ERRMAJ= .TRUE.
      ERRCAS= .FALSE.
      RETURN

C     MAKE THE CURVALINEAR DISTANCE CONTINUOUS
  800 DS     = 0.
      DO 805 I=2,NIM
      IF(S(I).EQ.0.) DS=S(I-1)
  805 S(I)   = S(I)+DS

C*** WRITE TOTAL COMPUTED DATA FOR THE BOUNDARY
      OMITFK= .TRUE.
      CALL FHEAD(NIM+4)
      WRITE (6,1800) (I,S(I),X(I),Y(I),ANGD(I),CURVB(I),FQK(I),I=1,NIM)
 1800 FORMAT(/21X24HCONSOLIDATED OUTPUT DATA//4X59HI      S          X,Z
     *       Y,R      ANGD     CURV      FQK/40X7HDEGREES/(2X,I3,0PF10.5
     *,2F11.5,F9.3,F10.6,F10.5,),)

      RETURN
```

87

```
1040 FORMAT(/1X59H***  ERROR - NUMBER OF INPUT POINTS (XA,YA) IS LESS T
    *HAN 2.)
1042 FORMAT(/1X34HINPUT TAPE RETRIEVAL INFORMATION -//2X7HFOUND =L3,)
1202 FORMAT(/8H SEGMENT,I3,9H  OF BDY=,A6/26H ------------------------
    *)
     END
```

```
*DECK SMOTH
      SUBROUTINE SMOTH
*SMOOTH          MAIN PROGRAM FOR SMOOTH                          -SMOOTH-

C     READ INPUT, DETERMINE NUMBER AND TYPE OF SEGMENTS
      CALL SMOINP

C     SMOOTH ARBITRARY SEGMENTS
      CALL SMOXEQ

C     CALC SPECIAL-CONTOUR SEGMENTS, WRITE OUTPUT
      CALL CONTRS

      RETURN
      END
```

```
*DECK SMOEXQ
      SUBROUTINE SMOXEQ
*SMOXEQ         ARBITRARY SEQMENT SMOOTHING                        -SMOXEQ-

      COMMON /CBITS / BITS,BLANK
      COMMON /CBEND / NBC(2),ANGE(2),CURVE(2),FEND(2)
      COMMON /CNTRL / K5(1),STA(2),INCLUD(2),DELETE(2),INSERT,CARRY
      EQUIVALENCE     (BDY,STA)
      COMMON /CSEGME/ IA(10),IB(10),IMA(10),IMB(10),JTYPE(10),
     1                N,NSEG, NII,NIM
       EQUIVALENCE    (NI,NII)
      COMMON /CSMOOA/ DEVA(20), FENDA(20),ANGA(20),CURVA(20), NARB
      COMMON /CSMOOB/ XA(100),YA(100),DEVI(100)
      COMMON /CDS2  / X(100),Y(100),ANG(100),ANGD(100),CURV(100),
     1                S(100),FQK(100),DEV(100),CURVB(100)
      COMMON /CLINES/ LINES,OMITFK,PTITLE(6)
       LOGICAL              OMITFK
      COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
       LOGICAL        ERR,ERRMAJ,INERR,PRERR
       LOGICAL        ERRCAS
       EQUIVALENCE    (ERRCAS,INERR)
      LOGICAL         DONE

C*** SMOOTH ARBITRARY CURVES
      NSWEEP= 1
  170 DONE  = .TRUE.
      ANGREF= 0.
      N     = 1
      NARB  = 1
  175 IF(JTYPE(N)-1) 189,176,190
  176 I     = IA(N)
      I2    = IB(N)
C      END CONDITIONS
      DEVI(1)=0.
      DEVI(12)=0.
      FEND(1)=0.
      FEND(2)=0.
      NBC(1)= 0
      NBC(2)= 0
      L     = 0
  180 LL    = NARB+20*L
      IF(FENDA(LL).EQ.BITS) GO TO 181
      NBC(1)= L
      FEND(1)=FENDA(LL)
  181 IF(FENDA(LL+1).EQ.BITS) GO TO 182
      NBC(2)= L
      FEND(2)=FENDA(LL+1)
  182 L     = L+1
      IF(L.LE.2) GO TO 180
C      CHECK FOR UNDEFINED END CONDITIONS
C       END-1
       IF(FEND(1).NE.999.) GO TO 184
       IF(N.EQ.1) GO TO 187
       IF(JTYPE(N-1).GE.0) GO TO 187
       IF(NBC(1).EQ.1) FEND(1)=ANGD(I-1)
       IF(NBC(1).EQ.2) FEND(1)=CURV(I-1)
C       END-2
  184  IF(FEND(2).NE.999.) GO TO 186
```

```
      IF(N.GE.NSEG) GO TO 200
      IF(JTYPE(N+1).GE.0) GO TO 187
      IF(NBC(2).EQ.1) FEND(2)=ANGD(I2+1)
      IF(NBC(2).EQ.2) FEND(2)=CURV(I2+1)
  186 IF(DEVA(NARB).NE.BITS) DEVI(I)=DEVA(NARB)
      IF(DEVA(NARB+1).NE.BITS) DEVI(I2)=DEVA(NARB+1)
      OMITFK= .TRUE.
      CALL FHEAD(17+I2-1)
      WRITE (6,1186) N,BDY
      S(I)  = 0.
      ANG(I)= ANGREF
      CALL SMOO
      JTYPE(N)=-1
      I2    = IB(N)
      ANGREF=ANG(I2)
      GO TO 188
  187 DONE  = .FALSE.
  188 IF(ERR) ERRCAS=.TRUE.
      ERR   = .FALSE.
  189 NARB  = NARB+2
  190 N     = N+1
      IF(N.LE.NSEG) GO TO 175

C     RETURN TO  170  TO LOOP THROUGH SEGMENTS AGAIN
C      TO PICK UP THOSE WHICH HAD UNDEFINED END CONDITIONS
      IF(DONE) RETURN
      NSWEEP= NSWEEP+1
      IF(NSWEEP.LE.10) GO TO 170
  200 WRITE (6,1200)
      ERRCAS=.TRUE.
      RETURN

 1186 FORMAT(/8H SEGMENT,I3,9H  OF BDY=,A6/26H ------------------------
     *)
 1200 FORMAT(1X50H***  ANGA,CURVA = 999  END OPTION USED INCORRECTLY)
      END
```

```
*DECK SMOO
      SUBROUTINE SMOO
*SMOO--          ANGLE, CURVATURE AND ARC LENGTH              -SMOO-
C                OF A SMOOTH CURVE PASSING CLOSE TO GIVEN POINTS
C                THE SMOOTHING OPTION HAS NOT BEEN INCLUDED.  INSTEAD, A
C                CURVE IS FITTED TO THE GIVEN X,Y POINTS.

C     INPUT-
C NA   MEANS NOT AVAILABLE IN THIS VERSION
C      IA,IB = RANGE OF INDEX IN LISTS XA,YA,DEVI,DEV,X,Y,ANG,CURV,E,S
C      XA    = LIST OF INPUT X
C      YA    = LIST OF INPUT Y
C NA   DEVI  = LIST OF POINT MOVEMENT PARAMETERS
C NA   TORQ1 = TORSIONAL SPRING COMPLIANCE - FIRST END
C NA   TORQN = TORSIONAL SPRING COMPLIANCE - SECOND END
C      NBC(L)= BOUNDARY CONDITION INDICATOR FOR FIRST(L=1) AND SECOND(L=2
C            = 0, 1, OR 2
C      ANGE(L)= ANGLE IN DEGREES, IF NBC(L)=1
C      CURVE(L)=CURVATURE,IF NBC(L)=2
C      FEND(L)=RATIO OF SHEAR FORCE,   END/NEXT TO END INTERVAL, IF NBC(L

C     NOTES-
C      THE UNITS OF XA,YA,DEVI,TORQ1 AND TORQN MUST BE THE SAME,
C      FOR EXAMPE, INCHES.  DEVI IS PROPORTIONAL TO THE CUBE ROOT OF
C      THE SPRING COMPLIANCES.  TORQS ARE DIRECTLYPROPORTIONALTOTHE
C      END TORSIONAL SPRING COMPLIANCES.  LARGER VALUES OF DEVI YEILD
C      LOWER APPLIED FORCES (AND GREATER DEVIATIONS), LARGER VALUES OF
C      TORQ YIELD LOWER APPLIED END MOMENTS.

C     OUTPUT BASED ON ADJUSTED POINTS-
C NA   DEV=V = DEVIATION FROM THE INPUT POINTS IN THE NORMAL DIRECTION,IN
C      X,Y   = ADJUSTED COORDINATES
C NA   ANG   = ANGLE IN RADIANS
C NA   ANGD  = ANGLE IN DEGREES
C NA   CURV  = CURVATURE, 1/IN
C NA   FQEI  = APPLIED FORCES, DELTA Y---, 1/IN2
C NA   S     = LENGTH ALONG THE CURVE, IN
C NA   ED    = ENERGY OF EQUIVALENT SPRINGS UNDER DEFLECTION DEV, 1/IN
C NA   ET    = SPRING ENERGIES, 1/IN
C NA   RMSDEV= ROOT MEAN SQUARE DEVIATION OF POINTS WITH DEVI.NE.0
C NA   RMSF  = ROOT MEAN SQUARE VALUE OF F/EI, 1/IN2
C NA   RMSF1 = ROOT MEAN SQUARE VALUE OF F/EI FOR UNADJUSTED BEAM

      COMMON /CCURV / NN,IDIM,G(2)
      COMMON /CB    / A(2)
       DIMENSION      U(2)
       EQUIVALENCE    (U,G)
      DIMENSION      V(100),W(100)
       EQUIVALENCE    (W,V)
      COMMON /CBEND / NBC(2),ANGE(2),CURVE(2),FEND(2)
       EQUIVALENCE    (NBC1,NBC),(NBC2,NBC(2))
      COMMON /CCUBE / NBCS(2),SAVS(4),FENDS(2)
      COMMON /CSEGME/ IIA(10),IIB(10),IMA(10),IMB(10),JTYPE(10),
     1                N,NSEG,NI,NIM
      COMMON /CSMOOB/ XA(100),YA(100),DEVI(100)
      COMMON /CDS2  / X(100),Y(100),ANG(100),ANGD(100),CURV(100),
     1                S(100),FQK(100),DEV(100),CURVB(100)
       DIMENSION      E(100)
```

```
      EQUIVALENCE    (E,FQK)
      COMMON /CSMOOD/ SGAMMA,SZETA1,SZETAN
      COMMON /ERASE / H(8,100)
      DIMENSION      CHD(8,99), G1(100),GN(100), INTER1(100)
      EQUIVALENCE    (CHD,H(8,1)), (INTER1,G1,H(1,1)), (GN,H(1,14))
      COMMON /CSMODE/ GAMMA(100)
      COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
      LOGICAL        ERR,ERRMAJ,INERR,PRERR

      DIMENSION      ENDPAR(3)
      DATA ENDPAR/5HFENDA,4HANGA,5HCURVA/

C     WRITE OUT END CONDITIONS
      ANGE(1)=FEND(1)
      ANGE(2)=FEND(2)
      CURVE(1)=FEND(1)
      CURVE(2)=FEND(2)
      WRITE (6,1020) ENDPAR(NBC1+1),FEND(1), ENDPAR(NBC2+1),FEND(2)
 1020 FORMAT(10X,47H* A CURVE HAS BEEN FITTED TO GIVEN X,Y POINTS *//
     1 6X,18HEND CONDITIONS -   , A5,4H(1)=,F9.5, 10X,A5,4H(2)=,F9.5)
     *H(2)=F9.5,)

      IA     = IIA(N)
      IB     = IIB(N)
      NPTS   = IB-IA+1
      IAB    = NPTS

C     CALC FORCES, F/EI, APPLIED TO THE BEAM WHICH PASSES THROUGH POINTS
      CALL BFACES(XA,YA,ANG,CURVB,E,S,IA,IB)
      CALL MOVE(2,XA(IA),X(IA),IAB,1, YA(IA),Y(IA),IAB,1)
      I      = IA
      K      = 1
  405 ANGD(I)=ANG(I)*57.29578
  415 K      = K+1
      I      = I+1
      IF(NPTS-K) 430,405,405
C  SMOOTHING LOGIC HAS BEEN REMOVED
  430 WRITE (6,1100)
      WRITE (6,1110) (XA(I),YA(I),DEVI(I),DEV(I),X(I),Y(I),ANGD(I),
     1               CURVB(I),FQK(I),S(I),I=IA,IB)
 1100 FORMAT(72X,10X,15HAPPLIED     ARC/6X17HINPUT COORDINATES17X,20HADJ
     *USTED COORDINATES22X,17HFORCES     LENGTH/7X89HXA,ZA       YA,RA
     * DEVI     DEV       X,Z          Y,R       ANGD      CURV       FQK
     *    S/33X,37H*1000                              DEGREES)
 1110 FORMAT(2X,2F11.5,F7.2,3PF7.2,0P2F11.5,F9.3,F10.6,2F10.5,)
      RETURN

      END
      OVERLAY(STC,1,2)
```

93

```
*DECK BUILDT
      PROGRAM BUILDT
      COMMON /CMAXIT/  MAXIT,MAJCTR,GREFIN,EDUM
      LOGICAL                        GREFIN
      COMMON /CPRINT/  PRTES2,PRTB,PRTA,PREFIN,PREFN2,SSONIC,PDUM(20)
      COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
       LOGICAL        ERR,ERRMAJ,INERR,PRERR
      COMMON /SELECT/ LENTRY

      GO TO (5,10,15) , LENTRY
    5 CALL BLDTAB
      GO TO 20
   10 CALL BPSORT
      MAJCTR= 0
C  INSERT SPECIAL BOUNDARY TYPES IN THE STATION TABLE
   15 CALL ISBOT
      IF(ERR) CALL ERROR1
      IF(PDUM(10).NE.0.) CALL EDUMP
   20 RETURN
      END
```

```
*DECK BLDTAB
      SUBROUTINE BLDTAB
*BLDTAB         COALLATE BDY-TABLE, BUILD LE-TE PT TABLE    -BLDTAB-

C     INPUT-
C      BOUNDARY TABLE, /BDYTAB/
C      CHANNEL INPUT DATE, /CHDATA/

C     OUTPUT-
C      CONDENSED BOUNDARY TABLE, /BDYTAB/
C      ORDERED EDGE POINTS, /LETEPT/

       COMMON /ALLCOM/ MACHA,PSA,TSA,PTA,TTA, AXIA,RGA,GAMA,
      1                MACHC,PSC,TSC,PTC,TTC, AXIC,RGC,GAMC,
      2                DAXIT,SCALEA,TTE,CHOTST
        REAL           MACHA(1),MACHC
        LOGICAL        AXIA,AXIC,CHOTST
C     BOUNDARY TABLE
C      INDEX- LB=LBDO,LBDE
C      LBNEXT= INCREMENT TO NEXT BOUNDARY
C      LBZ1   = INCREMENT TO THE FIRST BOUNDARY POINT (=0 BEFORE COALLATIO
C      CHNAME= CHANNEL WITH WHICH THE BOUNDARY DATA IS ASSOCIATED
C      UP     = T OR F FOR UPPER OR LOWER BOUNDARY
C      LEDEX = RELATIVE INDEX OF L.E. POINT WHEN LOWER AND UPPER SURFACE
C               CONTOURS ARE CONNECTED
C      BDNAME,LBA,LBB=NAME AND INDEX LIMITS OF SPECIFIC BOUNDARY
C                     DATA WHEN BOUNDARIES ARE COALLATED
       COMMON /CHDATA/ BDT(1),LBNEXT(1),LBZ1(1),
      1                CHNAME(1),UP(1),LEDEX(1),
      2                ZBT(1),RBT(1),ANGBT(42)
        LOGICAL        UP
        INTEGER BDT,CHNAME,BDNAME
        DIMENSION      BDNAME(1),LBA(1),LBB(1)
        EQUIVALENCE    (BDNAME,ZBT), (LBA,RBT), (LBB,ANGBT)
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
        DIMENSION      LIMITS(24)
        EQUIVALENCE    (LIMITS,LHO)
C     TABLE OF LEADING EDGE AND TRAILING EDGE POINTS
C      INDEX- LE=LEO,LEE,10
C      NLE,NTE=NO. OF L.E. AND T.E. COINCIDENT PTS, RESPECTIVELY
C      CHL,CHU=NAME OF CHANNEL ABOVE AND BELOW PT, RESPECTIVELY
C      BDL,BDU=BOUNDARY NAMES ASSOCIATED WITH THE POINTS
C      NUSED = COUNT OF TIMES THAT POINT USED IN CONSTRUCTION OF /ORTCHN/
       COMMON /LETEPT/ XE(1),YE(1),ANGE(1),NLE(1),NTE(1),
      1                CHL(1),CHU(1),BDL(1),BDU(1),NUSED(491)
        INTEGER        CHL,CHU,BDL,BDU

       COMMON /CBITS / BITS,IBLANK
       COMMON /CPI   / PI,TWOPI,PIQ2,PIQ4,TODEG,TORAD
       COMMON /ERASE / XX(1),YY,ANGG,NL,NT,CNL,CNU,BNL,BNU,NZERO
        DIMENSION      IXX(10)
        EQUIVALENCE    (IXX,XX)
        INTEGER        CNL,CNU,BNL,BNU
       COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
        LOGICAL        ERR,ERRMAJ,INERR,PRERR
```

```
      INTEGER     BD1,BD2,BNAME2,CHN,HLOWER,HUPPER,UPPER
      LOGICAL     WALL

      DATA HLOWER,HUPPER/5HLOWER,5HUPPER/

C     RELOCATE BDY-TABLE DOWN AND ADJACENT TO CHDATA-TABLE
      NMOVE = LBDE-LBDO+1
      CALL MOVE(1, BDT(LBDO),BDT(LHE+1),NMOVE,1)
      LBDO  = LHE+1
      LBDE  = LHE+NMOVE

C     DEFINE DOUBLE POINT TOLERANCE, DPTOL
      DPTOL = 1.E-5

C**   BOUNDARY TABLE SORT
C      RELOCATE TOGETHER THE BOUNDARIES WHICH BELONG TO THE SAME WALL
      LB1   = LBDO
  305 LB2   = LB1+LBNEXT(LB1)
      IF (LB2.GE.LBDE) GO TO 350
C      COMPARE CHANNEL NAME AND UPPER(LOWER) WALL
  310 IF(CHNAME(LB2).NE.CHNAME(LB1) .OR. (UP(LB2).AND. .NOT.UP(LB1))
     * .OR. (UP(LB1).AND. .NOT.UP(LB2)) ) GO TO 340
C      DOES LB2 FOLLOW LB1, COMPARE THE Z,R VALUES OF THE END POINTS
      L1    = LB1+LBNEXT(LB1)-9
      IF(ABS(ZBT(LB2)-ZBT(L1)).LT.DPTOL .AND.
     1   ABS(RBT(LB2)-RBT(L1)).LT.DPTOL) GO TO 315
C      DOES LB2 PRECEED LB1
      L2    = LB2+LBNEXT(LB2)-9
      IF(ABS(ZBT(L2)-ZBT(LB1)).GE.DPTOL .OR.
     1   ABS(RBT(L2)-RBT(LB1)).GE.DPTOL) GO TO 340
      LI    = LB1
      GO TO 316
  315 LI    = LB1+LBNEXT(LB1)
  316 NB2   = LBNEXT(LB2)
      L1    = LI+NB2
      L2    = LB2+NB2
      L22   = L2+NB2
      IF(LB2.EQ.LI)GO TO 340
      CALL MOVE(3, BDT(LI),BDT(LT),LI-1-LBDE,1,
     1             BDT(L2),BDT(LI),NB2,1,
     2             BDT(L22),BDT(L2),LBDE-L2+1,1)
      IF(LI.EQ.LB1) GO TO 305
  340 LB2   = LB2+LBNEXT(LB2)
      IF(LB2.LT.LBDE) GO TO 310
      LB1   = LB1+LBNEXT(LB1)
      GO TO 305

C**   COALLATE THE BOUNDARIES ALONG ONE WALL INTO ONE CONTOUR
  350 LB1   = LBDO
  355 NCOAL = 0
      CHN   = CHNAME(LB1)
      WALL  = UP(LB1)
  360 LB2   = LB1+LBNEXT(LB1)
      IF(LB2.GE.LBDE .OR. BDT(LB2).EQ.IBLANK) GO TO 400
C     IS THIS BOUNDARY CONTINUED
      IF(CHNAME(LB2).NE.CHN .OR. (UP(LB2).AND. .NOT.WALL) .OR.
     * (WALL .AND. .NOT.UP(LB2)) ) GO TO 380
```

```
      L1     = LB1+LBNEXT(LB1)-9
      L2     = LB2+LBZ1(LB2)
      IF(ABS(ZBT(L2)-ZBT(L1)).LT.DPTOL .AND.
     1   ABS(RBT(L2)-RBT(L1)).LT.DPTOL) GO TO 365
C       ERROR-  BOUNDARY TABLE NOT CONTINUOUS
      IUP=HLOWER
      IF( UP(LB1) ) IUP=UPPER
      WRITE(6,1365) IUP,CHNAME(LB1),ZBT(L1),RBT(L1),ZBT(L2),
     1                RBT(L2)
      CALL ERROR1

C     MOVE THE LB1 Z,R,ANG-DATA UP 6 SPACES IF THERE EXISTS
C      AN ANGLE DISCONTINUITY, 9 SPACES IF THERE DOES NOT.
C      (6 SPACES IS NOW ALWAYS USED SO THAT A PRIMARY ORTHOGONAL WILL BE
C      GENERATED AT BOUNDARY JUNCTIONS, 4/71)
  365 LUP    = 6
C      IF(ANGBT(L2).EQ.ANGBT(L1)) LUP=9
      LF     = LB1+6+LBZ1(LB1)
      LT     = LF+LUP
      NMOVE  = -((LB1+LBNEXT(LB1)) - LF)
      BNAME2= BDT(LB2)
      LNEXT2= LBNEXT(LB2)
      LSTART= LBZ1(LB2)

      CALL MOVE(1, BDT(LF),BDT(LT),NMOVE,1)

      IF(NCOAL.NE.0) GO TO 370
      NCOAL = 1
      BDNAME(LB1)=BDT(LB1)
      LBA(LB1)=LBZ1(LB1)
      LBB(LB1)=LBA(LB1)-NMOVE-3

  370 L1     = LB1+3*NCOAL
      BDNAME(L1)=BNAME2
      LBA(L1)=LBNEXT(LB1)
      LBB(L1)=LBA(L1) + (LNEXT2-(6+LSTART)) - 3
      N      = NCOAL
      NCOAL  = NCOAL+1

  375 IF(N.LE.0) GO TO 377
      L1     = LB1+3*(N-1)
      LBA(L1)=LBA(L1)+LUP
      LBB(L1)=LBB(L1)+LUP
      N      = N-1
      GO TO 375

  377 LBNEXT(LB1)=LBNEXT(LB1)+LNEXT2
      LBZ1(LB1)=LBZ1(LB1)+LUP
      GO TO 360

C     ELIMINATE GAPS
  380 IF(NCOAL.EQ.0) GO TO 390
      LDOWN = LBZ1(LB1) - 3*NCOAL
      IF(LDOWN.LE.0) GO TO 390
      LF     = LB1+6+LBZ1(LB1)
      LT     = LF-LDOWN
      NMOVE  = LBDE-LF+1
      CALL MOVE(1, BDT(LF),BDT(LT),NMOVE,1)
```

97

```
      LBNEXT(LB1)=LBNEXT(LB1)-LDOWN
      LBZ1(LB1)=LBZ1(LB1)-LDOWN
      N      = 1
  385 L1     = LB1+3*(N-1)
      LBA(L1)=LBA(L1)-LDOWN
      LBB(L1)=LBB(L1)-LDOWN
      N      = N+1
      IF(N.LE.NCOAL) GO TO 385
      LBDE   = LBDE-LDOWN

C     INDEX TO THE NEXT LB1
  390 LB1    = LB1+LBNEXT(LB1)
      IF(LB1.LT.LBDE) GO TO 355

*     INITIALIZE FAR FIELD INTERFACE BOUNDARY DATA IF REQ-D
  400 CALL FFINIT

C**   BUILD LEADING EDGE/TRAILING EDGE POINT TABLE,  /LETEPT/
      LEE    = LEO-1
      LB     = LBDO
  405 L1     = LB+LBZ1(LB)
      LL     = L1
      L2     = LB+LBNEXT(LB)-9
      GO TO 410
C      SEARCH FOR SHARP CORNERS
  407 LL     = LL+3
      IF(ABS(ZBT(LL)-ZBT(LL-3)).LT.DPTOL .AND.
     1   ABS(RBT(LL)-RBT(LL-3)).LT.DPTOL) GO TO 408
      IF(LL.LT.L2) GO TO 407
      GO TO 410
C      SHARP CORNER
  408 ZBT(LL)=ZBT(LL-3)
      RBT(LL)=RBT(LL-3)
      NZERO  = -1
      NL     = 0
      NT     = 0
      ANGG   = .5*(ANGBT(LL)+ANGBT(LL-3))
      GO TO 412
  410 NZERO  = 0
      ANGG   = ANGBT(LL)
  412 CALL SETM(1,IBLANK, CNL,4)
      XX     = ZBT(LL)
      YY     = RBT(LL)
      IF(UP(LB)) GO TO 415
C      LOWER BOUNDARY
      CNL    = CHNAME(LB)
      BNL    = BDT(LB)
      IF(LL.EQ.L1) GO TO 420
      IF(LL.EQ.L2) GO TO 425
      GO TO 435
C      UPPER BOUNDARY
  415 CNU    = CHNAME(LB)
      BNU    = BDT(LB)
      ANGG   = ANGG-PI
      IF(LL.EQ.L1) GO TO 425
      IF(LL.EQ.L2) GO TO 420
      GO TO 435
C      LEADING EDGE
```

```
  420 NL     = 1
      NT     = 0
      GO TO 435
C      TRAILING EDGE
  425 NT     = 1
      NL     = 0

C 435 CALL ESORTP

*ESORTP         PRELIMINARY EDGE POINT SORT                    -ESORT-******
C     SUBROUTINE ESORT

C    INPUT-
C     XX(10)= DATA VECTOR TO BE INSERTED INTO ARRAY-XE
C     XE     = ARRAY OF VECTORS SORTED ACCORDING TO FIRST TWO ELEMENTS
C     LEO,LEE=INDEX LIMITS OF THE XE-ARRAY

C    OUTPUT-
C     XE     = REVISED ARRAY OF EDGE POINTS
C     LEE    = REVISED UPPER LIMIT OF XE-ARRAY

C    SEARCH FOR ORDERED POSITION - J
  435 CONTINUE
      J      = 0
   55 I      = 1
   60 LE     = 10*J + I-1 + LEO
      IF(LE.GE.LEE) GO TO 80
      XD     = XX(I)-XE(LE)
      IF(ABS(XD).LE.(1.1*TTE)) XD=0.
      IF(XD) 80,70,65
   65 J      = J+1
      GO TO 55
   70 I      = I+1
      IF(I.LE.2) GO TO 60

C    THE NEW POINT IS COINCIDENT WITH POINT-J
      LE     = 10*J + LEO
      ANGE(LE)=.5*(ANGE(LE)+ANGG)
      NLE(LE)=NLE(LE)+NL
      NTE(LE)=NTE(LE)+NT
      I      = 6
   72 LE     = 10*J + I-1 + LEO
      IF(IXX(I).NE.IBLANK) XE(LE)=XX(I)
      I      = I+1
      IF(I.LE.10) GO TO 72
C     RETURN
      GO TO 436

C    RELOCATE AND INSERT THE NEW LINE IN LINE-J
   80 LEF    = 10*J + LEO
      LET    = LEF+10
      CALL MOVE(2, XE(LEF),XE(LET),LEF-LEE-1,1,
     1             XX,XE(LEF),10,1)
      LEE    = LEE+10
C     RETURN
C     END                                                  ************

  436 IF(LL-L2) 407,440,407
```

99

```
C       INCREMENT BOUNDARY TABLE INDEX
  440 LB     = LB+LBNEXT(LB)
        IF(LB.LT.LBDE) GO TO 405

C     CHECK FOR A MINIMUM OF  4  POINTS IN THE LETEPT-TABLE
        IF((LEE-LE0+1).LT.40) CALL ERROR1

C*    FINAL SORT OF /LETEPT/ BY AVERAGE FLOW ANGLE
        TANG   = 92./90.*PIQ2
        LE1    = LE0
  454 NCOUNT= (LEE+1-LE1)/10
  455 LE2    = LE1
  460 LE2    = LE2+10
        IF(LE2.GE.LEE) GO TO 470

C       IS PT2 IN FRONT OF PT1 (VECTOR PT1 TO PT2 GT 90 DEG FROM SL)
        ANGSL = .5*(ANGE(LE1)+ANGE(LE2))
        ANG12 = ATAN3(YE(LE2)-YE(LE1),XE(LE2)-XE(LE1),ANGSL)
        IF(ABS(ANG12-ANGSL).LE.TANG) GO TO 460

C       MOVE PT LE2 IN FRONT OF LE1
        LI     = LE1
        LT     = LI+10
        L2     = LE2+10
        L22    = L2+10
        CALL MOVE(3, XE(LI),XE(LT),LI-1-LEE,1,
       1             XE(L2),XE(LI),10,1,
       2             XE(L22),XE(L2),LEE-L2+1,1)
        NCOUNT= NCOUNT-1
        IF(NCOUNT.GE.0) GO TO 455
         WRITE (6,1468)
         CALL ERROR1

C     INDEX LE1
  470 LE1    = LE1+10
        IF(LE1.LT.LEE) GO TO 454

C*    COMBINE UPPER AND LOWER CONTOURS CONNECTED BY L.E. IN THE BDY-TABLE
C       LB1 AND LB2 ARE INDICIES OF THE TWO CONTOURS
C                        (LOWER AND UPPER SURFACE)
C       LUP    = ADDITIONAL SPACE REQD FOR SUBTABLE OF INCLUDED BOUNDARIES
        LE     = LE0
  472 IF(NLE(LE).NE.2) GO TO 496
        BD1    = BDU(LE)
        BD2    = BDL(LE)
        LB1    = LBF(BD1)
        LB2    = LBF(BD2)

C     CHECK L.E. ANGLE DISCREPANCY
        LB     = LB2+LBZ1(LB2)
        ANGD2  = ANGBT(LB)*TODEG
        NBD2   = BDT(LB2)
        LB     = LB1+LBNEXT(LB1)-9
        ANGD1  = ANGBT(LB)*TODEG
        NBD1   = BDT(LB1)
        IF(ABS(ANGD2-ANGD1).LT..1) GO TO 474
        ANGDAV= .5*(ANGD1+ANGD2)
```

```
      WRITE (6,1473) ZBT(LB ),RBT(LB ),NBD1,NBD2,ANGD1,ANGD2,ANGDAV
 1473 FORMAT (//52H ***  ERROR - THE BOUNDARY ANGLES AT L.E. POINT  Z =,
     1         F10.5,4H R =,F10.5//14X,17HARE NOT THE SAME.
     2         33H  THE AVERAGE VALUE WILL BE USED.//21X,7HBDY =  ,A6,6X,
     3         7HBDY =  ,A6/21X,5HANGD=,F8.3,6X,5HANGD=,F8.3//29X,
     4         8HAVG-ANG=,F8.3)
      ANGBT(LB)=ANGDAV*TORAD

C     MAKE ROOM FOR SUBTABLE OF INCLUDED BOUNDARIES
  474 LUP  = MAXO(3,LBZ1(LB1)) + MAXO(3,LBZ1(LB2)) - LBZ1(LB1)
      LB   = LB1+LBZ1(LB1)
      LT   = LB+LUP
      CALL MOVE(1, ZBT(LB),ZBT(LT),LB+5-LBDE,1)
      LBDE = LBDF+LUP
      IF(LB2.GE.LB1) LB2=LB2+LUP

C     INCLUDED BOUNDARIES IN COUNTOUR LB1
      IF(LBZ1(LB1).NE.0) GO TO 475
      BDNAME(LB)=BDT(LB)
      LBA(LB)= LUP
      LBB(LB)= LBA(LB)+LBNEXT(LB)-9
      LB   = LB+3
      GO TO 480
  475 LBN1 = LB1
  476 LBA(LBN1)=LBA(LBN1)+LUP
      LBB(LBN1)=LBB(LBN1)+LUP
      LBN1 = LBN1+3
      IF(LBN1.LT.LB) GO TO 476

C     UPPER SURFACE CHANNEL NAME IS STORED ON TOP OF -UP-
C     LEDEX = INDEX OF LEADING EDGE PT ON THE CONTOUR
  480 CHNAME(LB1+1)=CHNAME(LB2)
      LEDEX(LB1)=LBB(LB-3)

C     INCLUDED BOUNDARIES IN CONTOUR LB2
      IF(LBZ1(LB2).NE.0) GO TO 485
      BDNAME(LB)=BDT(LB2)
      LBA(LB)=LBB(LB-3)
      LBB(LB)=LBA(LB)+LBNEXT(LB2)-9
      GO TO 490
C      RELOCATE INDEX LIMITS OF UPPER BOUNDARIES
  485 LBN2 = LB2
      LBDIF = LBB(LB-3)-LBA(LB2)
  486 BDNAME(LB)=BDNAME(LBN2)
      LBA(LB)=LBA(LBN2)+LBDIF
      LBB(LB)=LBB(LBN2)+LBDIF
      LB   = LB+3
      LBN2 = LBN2+3
      IF(LBN2.LT.(LB2+LBZ1(LB2))) GO TO 486

C     RELOCATE LB2-COORDINATES INTO LB1-COUNTOUR.  NB2=NUMBER OF DATA
C     POINTS TO BE MOVED.
  490 NB2  = LBNEXT(LB2)-LBZ1(LB2)-9
      LI   = LB1+LBNEXT(LB1)+LUP
      LT   = LI+NB2
      L2   = LB2+LBZ1(LB2)+9
      L22  = LB2+LBNEXT(LB2)
      IF(LB2.LT.LB1) GO TO 494
```

101

```
      LB2    = LB2+NB2
      L2     = L2+NB2
      L22    = L22+NB2
  494 LBZ1(LB1)=LBZ1(LB1)+LUP
      LBNEXT(LB1)=LBNEXT(LB1)+LUP+NB2
      CALL MOVE(3, BDT(LI),BDT(LT),LI-1-LBDE,1,
     1             BDT(L2),BDT(LI),NB2,1,
     2             BDT(L22),BDT(LB2),LBDE+NB2+1-L22,1)
      LBDE   = LBDE+NB2-(L22-LB2)

      DO 495 LEX=LEO,LEE,10
  495 IF(BDL(LEX).EQ.BD2) BDL(LEX)=BD1
  496 LE     = LE+10
      IF(LE.LT.LEE) GO TO 472

      RETURN

 1468 FORMAT(/1X70HERROR- THE L.E., T.E. AND BOUNDARY POINTS CAN NOT BE
     *ORDERED ACCORDING/8X64HTO ORTHOGONAL NUMBER.  PLEASE CHECK S.L. AN
     *GLES IN TABLE-LETEPT.)
 1365 FORMAT(///1X8H**   THE3X,A6,1X25HBOUNDARY CONTOUR FOR CHN=A6,1X17H
     *IS NOT CONTINUOUS/6X9HAT POINTSF11.5,1H,F10.5,1X3HANDF11.5,1H,F10.
     *5,1H./6X59HTHE FOLLOWING TABLE CONTAINS THE BOUNDARY COORDINATE IN
     *PUT.)
      END
```

```
      Z(M1) = Z(M2)
      ILB(L2)=ISV
      FLB(L2)=FSV
      S1LB(L2)=SSV
      R(M2) = RSV
      Z(M2) = ZSV
      GO TO 100

C     COINCIDENT ORTHOGONALS
   85 M1     = MLB(L1)
      NAMBDY= NAMELB(L1)
      GO TO 187

C       UPPER BOUNDARY
  100 L2     = L1+LNEXT(L1)
  165 IF(L2.GE.LESTA) GO TO 190
      IF(NAMEUB(L2).EQ.NAMEUB(L1)) GO TO 170
      L2     = L2+LNEXT(L2)
      GO TO 165
C       NAME AGREEMENT
  170 IF(FLOAT(IUB(L1))+FUB(L1) - FLOAT(IUB(L2))-FUB(L2)) 180,185,190
C       SWITCH POINTS
  180 M1     = MUB(L1)
      M2     = MUB(L2)
      ISV    = IUB(L1)
      FSV    = FUB(L1)
      SSV    = S1UB(L1)
      RSV    = R(M1)
      ZSV    = Z(M1)
      IUB(L1)=IUB(L2)
      FUB(L1)=FUB(L2)
      S1UB(L1)=S1UB(L2)
      R(M1) = R(M2)
      Z(M1) = Z(M2)
      IUB(L2)=ISV
      FUB(L2)=FSV
      S1UB(L2)=SSV
      R(M2) = RSV
      Z(M2) = ZSV
      GO TO 190

C     COINCIDENT ORTHOGONALS
  185 M1     = MUB(L1)
      NAMBDY= NAMEUB(L1)
  187 ERR    = .TRUE.
      WRITE (6,1187) Z(M1),R(M1),NAMBDY

C     INDEX L1
  190 L1     = L1+LNEXT(L1)
      IF(L1.LT.LESTA) GO TO 60
      RETURN

 1187 FORMAT(45H ***  ERROR - COINCIDENT ORTHOGONALS AT POINT,2F10.5,11H
     * ALONG BDY=,A6)
      END
```

```
*DECK BPSORT
      SUBROUTINE BPSORT
*BPSORT         BOUNDARY POINT SORT                                  -BPSORT-

C     FIELD TABLES
C      INDEX- M=MO,NM
       COMMON /CZ     / Z(300)
       COMMON /CR     / R(300)
       COMMON /CS2    / S2(300)
       COMMON /CS1    / S1(300)
       COMMON /CPHI1  / PHI1(300)
       COMMON /CM     / JMS(300)
       COMMON /CCURV  / CURV(300)

       COMMON /CB     / B(300)
       COMMON /CIDEX  / M,J,MU,MD,ISTAG
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
        DIMENSION     LIMITS(24)
        EQUIVALENCE   (LIMITS,LHO)
       COMMON /SLTAB / W(128),X2(128),SLCHN(128)
        INTEGER SLCHN
       COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
      1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
      1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
      3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
        LOGICAL       PRIM
        INTEGER TYPELB,TYPEUB
        DIMENSION     SCHOKE(1)
        EQUIVALENCE   (SCHOKE,DWDV)
       COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
        LOGICAL       ERR,ERRMAJ,INERR,PRERR

C     BEGIN LOOP THROUGH STATION TABLE
       L1     = LO

C       LOWER BOUNDARY
   60  L2     = L1+LNEXT(L1)
   65  IF(L2.GE.LESTA) GO TO 100
       IF(NAMELB(L1).EQ.NAMELB(L2)) GO TO 70
       L2     = L2+LNEXT(L2)
       GO TO 65
C       NAME AGREEMENT
   70  IF(FLOAT(ILB(L2))+FLB(L2) - FLOAT(ILB(L1))-FLB(L1)) 80,85,100
C       SWITCH POINTS
   80  M1     = MLB(L1)
       M2     = MLB(L2)
       ISV    = ILB(L1)
       FSV    = FLB(L1)
       SSV    = S1LB(L1)
       RSV    = R(M1)
       ZSV    = Z(M1)
       ILB(L1)=ILB(L2)
       FLB(L1)=FLB(L2)
       S1LB(L1)=S1LB(L2)
       R(M1) = R(M2)
```

```
*DECK FFINIT
      SUBROUTINE FFINIT
*FFINIT         INITIALIZATION OF FAR FIELD CALC                -FFINIT-

      COMMON /CISBOT/ FARFLD(2),FREE(2),PRES(2),RFF,NZP,
     1                ZP(10),PPS(10), A1,A2,ADUM(6)
       INTEGER        FARFLD,FREE,PRES

      RETURN
      END
```

```
*DECK FRFDNZ
      SUBROUTINE FRFDNZ
CFRFDNZ         GENERATE ZDN, ZIJ MATRIX FOR FAR-FIELD BC.        -FRFDNZ-
      COMMON /ALLCOM/ MACHA,PSA,TSA,PTA,TTA, AXIA,RGA,GAMA,
     1                MACHC,PSC,TSC,PTC,TTC, AXIC,RGC,GAMC,
     2                DAXIT,SCALEA,TTE,CHOTST
        REAL          MACHA(1),MACHC
        LOGICAL       AXIA,AXIC
        LOGICAL       CHOTST
      COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
     *                LO,LESTA,  LDUM(8),
     *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
     *                LEO,LEE, LRO,LRE,LRD
        DIMENSION     LIMITS(24)
        EQUIVALENCE   (LIMITS,LHO)
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
      COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
     1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
     1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
     3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
        LOGICAL       PRIM
        INTEGER TYPELB,TYPEUB
        DIMENSION     SCHOKE(1)
        EQUIVALENCE   (SCHOKE,DWDV)
      COMMON /ERASE2/ WSTA(100),DISP(100),WAKE(100),TT(100),PT(100),
     *                LAM(100),RGX(100),C2CPX(100),DUM(534),
     *                IN1(25),IN2(25,2),
     *                NINT,M(21),EE(21),KK(21),XINT(21),
     *                YINT(21),ZZ(21)
      REAL        M    ,   KK
      DIMENSION   XIJ(25,25),YIJ(25,25)
      EQUIVALENCE (WSTA,XIJ),(C2CPX,YIJ)
      COMMON /ERASE/ UNIT(25,25)
      COMMON  /CPI    / PI,DUMPI(5)
      COMMON  /CBITS /  BITS,BLANK
      COMMON /CFRFLD/ NFF,MAXFF,ZFF(64),RFF(64),
     *                           ZDN(25),DRDN(25),UDN(25),ZIJ(25,25)
      DIMENSION     FGRX(100)
      EQUIVALENCE   (FGRX,ZIJ)
      COMMON /CFRFIN/ ATINF,MINF,RFFREF,UINF,ZDN1,ZDN25
      REAL                  MINF
      COMMON /CPRINT/ PDUM(26)
      COMMON /CISBOT/ DUMIS(30),ADUM(6)
      COMMON /CPTMOV/ VELPOT,ICOB,NODENS,CPTDUM
      COMMON /CR    / R(300)
      EQUIVALENCE (R1,RFFREF),(R25,ADUM(2))
      LOGICAL   DSIZE
      REAL  M1,M2,M3,M4
      DATA AK1,AK2,AK3,AK4,AK5/
     * 1.3862944,.096663443,.035900924,.037425637,.0145119621/
      DATA BK1,BK2,BK3,BK4,BK5/
     * .5,.12498594,.06880249,.03328355,.00441787/
      DATA AE1,AE2,AE3,AE4/
     * .44325141,.06260601,.04757384,.01736506/
```

```
      DATA BE1,BE2,BE3,BE4/
     * .24998368,.0920018,.04069698,.005264496/
C*
C
C     INPUT***
C      MINF       = FREE STREAM MACH NUMBER .
C      ZDN1,ZDN25= STREAMWISE LIMITS OF FAR FIELD .
C      RFFREF     = NOMINAL RADIUS OF FAR FIELD .
C     OUTPUT***
C      ZDN(1-25) = STREAMWISE CO-ORDINATES FOR DN- FAR FIELD SOLUTION .
C      ZIJ(25,25)= Z MATRIX = (INVERSE OF YIJ)*XIJ-
C     EXTENSION FRACTION TO FF= ADUM(1)

      BETA  = SQRT(1.-MINF**2)
      OBETA = 1./BETA
C     INITIALIZE DZ, ZDN TABLE
C     TRANSFORM TO INCOMPRESSIBLE PLANE
      NDENSV= NODENS
      NODENS= -1
    1 DZFF  = ZDN25-ZDN1
      ZDN1  = ZDN1-ADUM(1)*DZFF
      ZDN25 = ZDN25+ADUM(1)*DZFF
      ZDN(1)= ZDN1*OBETA
      ZDN(25)= ZDN25*OBETA
      DZ    = (ZDN(25)-ZDN(1))/24.
      DO 2 K=2,24
    2 ZDN(K)= ZDN(K-1)+DZ
C     DETERMINE FF CROSS STREAM COORDINATE AT ZDN(25)
      L     = LESTA-19
      IF( LNEXT(L).NE.20 ) CALL ERROR1
      MA    = MLB(L)
      MB    = MUB(L)
      CALL TTPT(MA,MB,WSTA,DISP,WAKE,TT,PT,LAM,RGX,C2CPX,FGRX)
      NK    = MB-MA+1
C     ASSUME ISENTROPIC PROCESS TO UNDISTURBED CONDITIONS AT ZDN(25)
      GM2   = .5*(GAMA-1.)
      GM1   = (GAMA-1.)/GAMA
      PSINF = PT(NK)/(1.+GM2*MINF**2)**(1./GM1)
      AREA  = 0.
      K     = 0
 1111 K     = K+1
      GMA   = (1.+FGRX(K))/FGRX(K)
      GM1   = 1./(GMA*FGRX(K))
      TSQTT = (PSINF/PT(K))**GM1
      V2    = SQRT(C2CPX(K)*TT(K)*(1.-TSQTT))
      RHO2  = PT(K)/(RGX(K)*TT(K))*TSQTT**FGRX(K)
      IF( K.GT.1 ) GO TO 1112
      WQAKM1= RHO2*V2
      GO TO 1111
 1112 WQA   = RHO2*V2
      AREA  = AREA+2.*(WSTA(K)-WSTA(K-1))/(WQAKM1+WQA)
      WQAKM1= WQA
      IF( K.LT.NK ) GO TO 1111
      R25   = AREA+R(MA)
      IF( AXIA ) R25=SQRT(R(MA)**2+AREA/PI)
      IF( .NOT.AXIA ) GO TO 94
      NINT  = 11
```

```
C     INITIALIZE PARAMETERS FOR INTEGRATION

    3 DZZ    = DZ/FLOAT(NINT-1)
C     NOTE**** RADIAL CO-ORDINATE SCALED******
      DSING = 0.1*RFFREF
      DSIZE = .TRUE.
      IF( DZZ.LE.DSING ) DSIZE=.FALSE.
      FA     = 4.*RFFREF**2
      IF( DSIZE ) DELZD=DZZ-DSING
      DD     = AMIN1( DZZ,DSING )
      AL     = ALOG( .125*DD )
      SINGV = 2.*(-PI+DD*AL)-.125*DD**3*(1.+AL)

C     OUTER LOOP FOR CALC. OF XIJ,YIJ TABLES
      DO 90 I=1,25
C     INNER LOOP FOR CALC. OF XIJ,YIJ TABLES
      DO 89 J=1,25
C     SECTION TO BUILD TABLES FOR INTEGRATION
C     TABLES ARE BUILT IN 2 PASSES
      KGO    = 1
      IF( I.EQ.J ) GO TO 10
      IF( J.EQ.1 ) KGO=2
      IF( J.EQ.25) KGO=3
      GO TO 12
   10 KGO    = 4
      IF( J.EQ.1 ) KGO=5
      IF( J.EQ.25) KGO=6
   12 NIN    = NINT
      IF( KGO.NE.1 .AND. KGO.NE.4 ) NIN=(NINT-1)/2+1
      NMID=0
      IF( KGO.EQ.4 ) NMID=(NINT-1)/2+1

C     INITIAL PASS TO BUILD TABLES

      K      = 0
   15 K      = K+1
      K1     = K-1
      C      = 1.
      GO TO (20,25,20,30,35,40) , KGO

C     NORMAL BRANCH--OR  (J=25, I.NE.J )
   20 IF( K.GT.1 ) GO TO 22
   21 ZZ(K) = ZDN(J)-.5*DZ
      GO TO 23
   22 ZZ(K) = ZZ(K1)+C*DZZ
   23 M(K)  = FA/(FA+(ZDN(I)-ZZ(K))**2)
      GO TO 50
C     **(J=1,I.NEJ)
   25 IF( K.GT.1 ) GO TO 22
      ZZ(K) = ZDN(1)
      GO TO 23
C     NORMAL SINGULARITY BRANCH
   30 IF( K.EQ.1 ) GO TO 21
      IF( ZZ(K-1).NE.BITS ) GO TO 31
      K1     = K-2
      C      = 2.
   31 IF( K.NE.NMID ) GO TO 22
   32 ZZ(K) = BITS
```

```
      M(K)  = BITS
      GO TO 50
C    **(I=J,J=1)
   35 IF( K.GT.2 ) GO TO 22
      GO TO (32,36) , K
   36 ZZ(K) = ZDN(J)+DZZ
      GO TO 23
C    **(I=J,J=25)
   40 IF( K.EQ.1 ) GO TO 21
      IF( K.EQ.NIN ) GO TO 32
      GO TO 22
   50 IF( K.LT.NIN ) GO TO 15

C    FINAL PASS TO BUILD TABLES-- ADJUST FOR  SINGULARITIES CLOSER
C    THAN  DZZ

      IF( .NOT. DSIZE ) GO TO 70
      K     = 0
   55 K     = K+1
      IF( ZZ(K).NE.BITS ) GO TO 60
      GO TO (60,60,60,56,57,56) , KGO
   56 ZZ(K-1)= ZZ(K-1)+DELZD
      M(K-1)= FA/(FA+(ZDN(I)-ZZ(K-1))**2)
      IF( KGO.EQ.6 ) GO TO 60
   57 ZZ(K+1)= ZZ(K+1)-DELZD
      M(K+1)= FA/(FA+(ZDN(I)-ZZ(K+1))**2)
   60 IF( K.LT.NIN ) GO TO 55

C    EVALUATE ELLIPTIC INTEGRALS  (K(M),E(M))

   70 DO 71 L=1,NIN
      IF( M(L).EQ.BITS ) GO TO 71
      M1    = 1.-M(L)
      IF( M1.EQ.1. .OR. M1.EQ.0. ) CALL ERROR1
      M2    = M1*M1
      M3    = M2*M1
      M4    = M2*M2
      TLOG  = ALOG(1./M1)
C
C     EVALUATE KK
C
      KK(L) = AK1+AK2*M1+AK3*M2+AK4*M3+AK5*M4
     *       +(BK1+BK2*M1+BK3*M2+BK4*M3+BK5*M4) *TLOG
C
C     EVALUATE EE
C
      EE(L) = 1.+AE1*M1+AE2*M2+AE3*M3+AE4*M4
     *       +(BE1*M1+BE2*M2+BE3*M3+BE4*M4) *TLOG
C
   71 CONTINUE

C    CALCULATE INTEGRANDS  XINT,Y-NT

      DO 73 K=1,NIN
      IF( ZZ(K).EQ.BITS ) GO TO 73
      DEN   = SQRT(FA+(ZDN(I)-ZZ(K))**2)
      XINT(K)= -4.*RFFREF*EE(K)/(DEN*(ZDN(I)-ZZ(K)))
      YINT(K)= -2.*(KK(K)-EE(K))/DEN
```

```
   73 CONTINUE
C     INTEGRATE
   75 XIJI   = 0.
      YIJI   = 0.
      K      = 1
   76 K      = K+1
      GO TO (77,77,77,78,78,78) , KGO
   77 DZK    = ZZ(K)-ZZ(K-1)
      TERMX  = 0.5*(XINT(K)+XINT(K-1))
      TERMY  = 0.5*(YINT(K)+YINT(K-1))
      XIJI   = XIJI+TERMX*DZK
      YIJI   = YIJI+TERMY*DZK
      GO TO 80
C
   78 IF( (ZZ(K).NE.BITS) .AND. (ZZ(K-1).NE.BITS) ) GO TO 77
      IF( KGO.EQ.6 ) GO TO 80
      IF( KGO.EQ.4 ) K=K+2
      IF( KGO.EQ.5 ) K=K+1
      GO TO 77
C
   80 IF( K.LT.NIN ) GO TO 76
      XIJ(I,J)= XIJI
      IF( KGO.GT.3 ) YIJI=YIJI+SINGV
      YIJ(I,J)= YIJI
   89 CONTINUE
   90 CONTINUE
      IF( PDUM(26).EQ.0. ) GO TO 91
      CALL TABPRT(3HXIJ,XIJ,625,10)
      CALL TABPRT(3HYIJ,YIJ,625,10)
   91 CONTINUE
C     DETERMINE INVERSE OF YIJ
      CALL MATINV(YIJ,25,UNIT,0,DET,IN1,IN2,25,ISCALE)
      DO 93 I=1,25
      DO 93 J=1,25
      ZIJ(I,J)= 0.
      DO 92 K=1,25
   92 ZIJ(I,J)= ZIJ(I,J)+XIJ(I,K)*UNIT(K,J)
   93 CONTINUE
C     TRANSFORM BACK TO COMPRESSIBLE PLANE
      GO TO 97
   94 CALL SETM(1,0.,ZIJ,625)
      DO 96 I=1,25
      DO 95 J=1,25
      IF(I.EQ.J) GO TO 95
      DXIJP = ZDN(I)-(ZDN(J)+.5*DZ)
      DXIJM = ZDN(I)-(ZDN(J)-.5*DZ)
      ZIJ(I,J)= -1./PI*ALOG(DXIJP/DXIJM)
   95 CONTINUE
   96 CONTINUE
   97 CALL FMPYC(1, BETA,ZDN,ZDN,25)
      CALL FMPYC(1,OBETA,ZIJ,ZIJ,625)
      IF( PDUM(26).EQ.0. ) GO TO 200
      CALL TABPRT(5HYIJ-1,UNIT,625,10)
```

```
      CALL TABPRT(3HZIJ,ZIJ,625,10)
  200 NODENS= NDENSV
      WRITE (6,211) ZDN1,R1,ZDN25,R25
  211 FORMAT(//6X,29H*EXTENDED FAR FIELD BOUNDARY*/7X,2HZ=,F10.3,3X,
     * 2HR=,F10.3/7X,2HZ=,F10.3,3X,2HR=,F10.3/)
      RETURN
      END
```

```
*DECK ISBOT
      SUBROUTINE ISBOT
*ISBOT-          INSERT SPECIAL BOUNDARY TYPES                    -ISBOT-

C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
       COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
      1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
      1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1U8(1),
      3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
        LOGICAL        PRIM
        INTEGER TYPELB,TYPEUB
        DIMENSION      SCHOKE(1)
        EQUIVALENCE    (SCHOKE,DWDV)
C     BOUNDARY TABLE
C      INDEX- LB=LBDO,LBDE
C      LBNEXT= INCREMENT TO NEXT BOUNDARY
C      LBZ1   = INCREMENT TO THE FIRST BOUNDARY POINT (=0 BEFORE COALLATIO
C      CHNAME= CHANNEL WITH WHICH THE BOUNDARY DATA IS ASSOCIATED
C      UP     = T OR F FOR UPPER OR LOWER BOUNDARY
C      LEDEX = RELATIVE INDEX OF L.E. POINT WHEN LOWER AND UPPER SURFACE
C              CONTOURS ARE CONNECTED
C      BDNAME,LBA,LBB=NAME AND INDEX LIMITS OF SPECIFIC BOUNDARY
C                    DATA WHEN BOUNDARIES ARE COALLATED
        DIMENSION      BDT(1),LBNEXT(1),LBZ1(1),
      1                CHNAME(1),UP(1),LEDEX(1),
      2                ZBT(1),RBT(1),ANGBT(42)
        LOGICAL        UP
        INTEGER BDT,CHNAME,BDNAME
        DIMENSION      BDNAME(1),LBA(1),LBB(1)

        DIMENSION      CHNAM(1),LHNEXT(1)
        INTEGER         CHNAM
        EQUIVALENCE    (X1,BDT,CHNAM), (LNEXT,LBNEXT,LHNEXT), (MLB,LBZ1),
      1                (MUB,CHNAME), (PRIM,UP), (TYPELB,LEDEX),
      2                (NAMELB,ZBT,BDNAME), (ILB,RBT,LBA), (FLB,ANGBT,
      3                LBB)

       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
        DIMENSION      LIMITS(24)
        EQUIVALENCE    (LIMITS,LHO)

       COMMON /CFRFIN/ ATINF,MINF,RFFREF,UINF,ZDN1,ZDN25
       REAL                  MINF
       COMMON /CIDEX / M,J,MU,MD,ISTAG
       COMMON /CISBOT/ FARFLD(2),FREE(2),PRES(2),RFF,NZP,
      1                ZP(10),PPS(10), A1,A2,ADUM(6)
        INTEGER        FARFLD,FREE,PRES
       COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
        LOGICAL        ERR,ERRMAJ,INERR,PRERR
```

```
*DECK BFAS
      SUBROUTINE BFAS(X,Y,ANG,S,KA,KB)
*BFAS--          BEAM FIT EVALUATION OF ANGLE AND S              -BFAS-
      DIMENSION X(10),Y(10),ANG(10),S(10)

C     INPUT-
C      X,Y    - COORDINATES
C      ANG    - ANGLE IN RADIANS  (IF MA=1)
C      ANG(1)= ESTIMATED ANGLE AT THE FIRST POINT (MA=0)
C      KA,KB  - FIRST AND LAST INDEX OF VARIABLES  X,Y,ANG,CURV,E AND S
C      KD     - STORAGE INCREMENT OF  X,Y,ANG,CURV,E, AND S
C      KORDER= 0 IF ERROR1 IS TO BE CALLED WHEN PTS ARE OUT OF ORDER
C            = 1 IF RETURN IS TO BE MADE FOR CORRECTIVE ACTION
C            = -1 IF POINT ORDER CHECK IS TO BE SKIPPED

C     OUTPUT-
C      ANG    - ANGLE IN RADIANS
C      S      - ARC LENGTH ALONG THE CURVE,  (L)
C      KORDER= INDEX OF 2ND OF ADJACENT OUT-OF-ORDER PTS (=1 ON ENTRY).

      COMMON /CBEAM / MA,MB,KD,KORDER
      COMMON /ERASE / A(3),B(1),YPB(1),DA(1),ACHD(1),CHD(793)

      NK     = KB

      CALL BEAM(X(KA),Y(KA),ANG(KA),(KB-KA+KD)/KD)
      IF(KORDER.NE.0) GO TO 800

C     (K=KA)
      SK     = S(KA)

C     (K=KA+1,KB)
      I      = 9
      K      = KA+KD
   70 SK     = SK + CHD(I-8)*(1.+(B(I-8)*B(I-8)-.5*B(I-8)*YPB(I-8)+
     1          YPB(I-8)*YPB(I-8))/15.)
      S(K)   = SK
      IF(K-NK) 80,900,900
   80 I      = I+8
      K      = K+KD
      GO TO 70

C     OUT OF ORDER POINTS
  800 KORDER= KA+KORDER-KD

  900 RETURN
      END
```

```
*DECK FARFLD
      SUBROUTINE FARFLD
CFARFLD         COMPUTATION OF VELOCITY ON FAR FIELD BOUNDARY    -FARFLD-

       COMMON /CR     / R(300)
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LFO,LEE, LRO,LRE,LRD
        DIMENSION      LIMITS(24)
        EQUIVALENCE    (LIMITS,LHO)
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
       COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
      1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
      1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
      3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
        LOGICAL        PRIM
        INTEGER TYPELB,TYPEUB
        DIMENSION      SCHOKE(1)
        EQUIVALENCE    (SCHOKE,DWDV)
       COMMON /CZ     / Z(300)
       COMMON /CFRFIN/ ATINF,MINF,RFFREF,UINF,ZDN1,ZDN25
       COMMON /CFRFLD/ NFF,MAXFF,ZFF(64),RFF(64),
      *                          ZDN(25),DRDN(25),UDN(25),ZIJ(25,25)
       COMMON /CIDEX / M,J,MU,MD,ISTAG
       COMMON /CPHI1 / PHI1(300)
       COMMON /CPRINT/ PDDUM(16),PREF,PREFD,PREFI,PDDDUM(7)
       COMMON /ERASE / EDUM(711),PHIFF(64),RDN(25)
       COMMON /CISBOT/ DUMIS(30),ADUM(6)
       EQUIVALENCE (R1,RFFREF),(R25,ADUM(2))
       EQUIVALENCE (Z1,ZDN1),(Z25,ZDN25)
C     INPUT***
C      FIELD TABLES R,Z
C      VALUES OF M ON OUTER STREAMLINE
C      Z MATRIX FROM DN SOLUTION OF FAR FIELD
C     OUTPUT***
C      TABLE OF UDN VS ZDN
C     PREFI=0 USE LFIT1(NORMAL)     PREFI=1 USE LSPFIT -----FROM PHI1
C
C     GET R,Z VALUES FROM FIELD TABLES (OUTER STREAMLINE)
C
       L      = LO
     1 M      = MBEGIN(NJ)
       CALL STAND(M,L,UPPER)
       DATA KFAR/6HFARFLD/
       IF( TYPEUB(L).NE.KFAR    ) RETURN
       NF     = 0
     2 NF     = NF+1
       RFF(NF)= R(M)
       ZFF(NF)= Z(M)
       PHIFF(NF)= PHI1(M)
       CALL GETIX
       M      = MD
       IF( M.NE.0 ) GO TO 2
```

```
*DECK BFAC
      SUBROUTINE BFAC(X,Y,ANG,CURV,NK)
*BFAC--          BEAM FIT EVALUATION OF ANGLE, CURVATURE      -BFAC-
      DIMENSION X(10),Y(10),ANG(10),CURV(10)

C     INPUT-
C      X,Y    - COORDINATES
C      ANG    - ANGLE IN RADIANS  (IF MA=1)
C      NK     = LENGTH OF X,Y,ANG,CURV-LISTS

C     OUTPUT-
C      ANG    - ANGLE IN RADIANS
C      CURV   - CURVATURE

       COMMON /CBEAM / MA,MB,KD
       COMMON /ERASE / A(3),B(1),YPB(1),DA(1),ACHD(1),CHD(793)

       CALL BEAM(X,Y,ANG,NK)

       I      = 1
C      KA     = 1
       KB     = (NK-1)*KD+1
       K      = 1
C      (K=KA,KB-1)
   60 CURV(K)= (4.*B(I)+2.*YPB(I))/(CHD(I)*(1.+1.5*B(I)*B(I)))
   80 I      = I+8
      K      = K+KD
      IF(K-KB) 60,90,90

C      (K=KB)
   90 CURV(K)=(-2.*B(I-8)-4.*YPB(I-8))/(CHD(I-8)*(1.+1.5*YPB(I-8)*YPB(I-
     1          8)))

       RETURN
       END
```

```
*DECK BFACS
      SUBROUTINE BFACS(X,Y,ANG,CURV,S,KA,KB)
*BFACS-         BEAM FIT EVALUATION OF ANGLE, CURVATURE,     -BFACS-
C                              AND  S
      DIMENSION X(10),Y(10),ANG(10),CURV(10),S(10)

C     INPUT-
C      X,Y    - COORDINATES
C      ANG    - ANGLE IN RADIANS  (IF MA=1)
C      ANG(1)= ESTIMATED ANGLE AT THE FIRST POINT (MA=0)
C      KA,KB  - FIRST AND LAST INDEX OF VARIABLES  X,Y,ANG,CURV,E AND S
C      KD     - STORAGE INCREMENT OF  X,Y,ANG,CURV,E, AND S

C     OUTPUT-
C      ANG    - ANGLE IN RADIANS
C      CURV   - CURVATURE
C      S      - ARC LENGTH ALONG THE CURVE,  (L)

      COMMON /CBEAM / MA,MB,KD
      COMMON /ERASE / A(3),B(1),YPB(1),DA(1),ACHD(1),CHD(793)

      NK     = KB

      CALL BFAS(X,Y,ANG,S,KA,KB)

      I      = 1
      K      = KA
C     (K=KA,KB-1)
   60 CURV(K)= (4.*B(I)+2.*YPB(I))/(CHD(I)*(1.+1.5*B(I)*B(I)))
   80 I      = I+8
      K      = K+KD
      IF(K-NK) 60,90,90

C     (K=KB)
   90 CURV(K)=(-2.*B(I-8)-4.*YPB(I-8))/(CHD(I-8)*(1.+1.5*YPB(I-8)*YPB(I-
     1          8)))

      RETURN
      END
```

```
CALL BFI
ZD     = ZBT(I)+ZM
RD     = RBT(I)+RM
ANGD   = ANGCHD+ANGM
CURVD  = CURVM
SIDD   = SIM

FD     = F
INTVL  = (I - (LB+LBZ1(LB)))/3 + 1

RETURN
END
```

```
*DECK BF3
      SUBROUTINE BF3(X,Y,ANG,CURV, IA,IB)
*BF3              CENTRAL 3-POINT CURVATURE                          -BF3-
       DIMENSION       X(10),Y(10),ANG(10),CURV(10)
      COMMON /CBEND / NBCB(2),ANGE(2),CURVE(2),FB(2)
      DIMENSION       ANGX(3),CURX(3)
      NBCB(1)=0
      NBCB(2)=0
      IBM2   = IB-2
      ANGX(1)=0.
      IF( IBM2.LT.IA ) RETURN
      DO 110 I=IA,IBM2
      CALL BFAC(X(I),Y(I),ANGX,CURX,3)
      ANG(I+1)=ANGX(2)
  110 CURV(I+1)=CURX(2)
      RETURN
      END
```

```
*DECK BDYPTM
      SUBROUTINE BDYPTM(NAME,INTVL, ZD,RD,FD,SIDD,DS1,DSIGMA)
*BDYPTM          BOUNDARY POINT MOVEMENT                          -BDYPTM-

C     INPUT-
C      BDT    = BOUNDARY TABLE
C      NAME   = BOUNDARY NAME
C      INTVL  = INDEX OF INTERVAL OF THE INPUT POINT IN THE BOUNDARY TABLE
C      FD     = FRACTION POSITION OF THE INPUT POINT IN THE INTERVAL
C      SIDD   = ARC DISTANCE FROM THE BEGINING OF THE INPUT INTERVAL
C      DS1    = REQ-D MOVEMENT IN THE CLOCKWISE DIRECTION FROM THE INPUT P

C     OUTPUT-
C      INTVL  = INDEX OF INTERVAL OF THE OUTPUT POINT
C      ZD,RD  = COORDINATES OF THE CALCULATED OUTPUT POINT
C      ANGD   = ANGLE OF OUTPUT POINT
C      CURVD  = CURVATURE OF OUTPUT POINT
C      FD     = FRACTION POSITION IN THE OUTPUT INTERVAL
C      SIDD   = ARC DISTANCE FROM THE BEGINING OF THE OUTPUT INTERVAL
C      DSIGMA= -GET- MINUS -ASK- POINT MOVEMENT DISTANCE

C     BOUNDARY TABLE
C      INDEX- LB=LBDO,LBDE
C      LBNEXT= INCREMENT TO NEXT BOUNDARY
C      LBZ1   = INCREMENT TO THE FIRST BOUNDARY POINT (=0 BEFORE COALLATIO
C      CHNAME= CHANNEL WITH WHICH THE BOUNDARY DATA IS ASSOCIATED
C      UP     = T OR F FOR UPPER OR LOWER BOUNDARY
C      LEDEX  = RELATIVE INDEX OF L.E. POINT WHEN LOWER AND UPPER SURFACE
C               CONTOURS ARE CONNECTED
C      BDNAME,LBA,LBB=NAME AND INDEX LIMITS OF SPECIFIC BOUNDARY
C                     DATA WHEN BOUNDARIES ARE COALLATED
       COMMON /CHDATA/ BDT(1),LBNEXT(1),LBZ1(1),
      1                CHNAME(1),UP(1),LEDEX(1),
      2                ZBT(1),RBT(1),ANGBT(42)
         LOGICAL       UP
         INTEGER BDT,CHNAME,BDNAME
         DIMENSION     BDNAME(1),LBA(1),LBB(1)
         EQUIVALENCE   (BDNAME,ZBT), (LBA,RBT), (LBB,ANGBT)
       COMMON /CBEAM2/ DR,DZ,YPA,YPB,F,G, DX,YQDX,ZM,RM,ANGM,CURVM,SIM,
      1                RZONLY, ANGCHD,SINTVL, YPASQ,YPAB,YPBSQ
         LOGICAL       RZONLY
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
         DIMENSION     LIMITS(24)
         EQUIVALENCE   (LIMITS,LHO)
       COMMON /CBDYPT/ ANGD,CURVD
       COMMON /CBITS / BITS,BLANK

       F      = FD
       SID    = SIDD
       IF(F.EQ.0. .OR. F.EQ.1.) F=BITS
       DSIGMA= 0.

C     SEARCH FOR MATCHING BOUNDARY NAME
       LB     = LBF(NAME)
       IF(LB.EQ.0) CALL ERROR1
```

```
C       I       = INDEX OF POINT WHICH BEGINS THE INTERVAL
C       SFI     = DISTANCE FROM POINT (I)
C       SFIP1   = DISTANCE FROM POINT (I+1)
        MINI    = LB+LBZ1(LB)
        I       = MINI+3*(INTVL-1)
        MAXI    = LB+LBNEXT(LB)-12
    75  CALL BARC(I)
C           IF -I- IS THE FIRST OF A DOUBLE POINT, BACK UP TO PREV INTERVAL
            IF(SINTVL.NE.0.) GO TO 80
            I       = I-3
            FD      = 1.
            IF(I.LT.MINI) CALL ERROR1
            GO TO 75
    80  IF(FD.EQ.1. .OR. SID.GT.SINTVL) SID=SINTVL
        SFI     = DSI+SID
        SFIP1   = SFI-SINTVL

C     IS THE NEW POINT WITHIN THIS INTERVAL
   100  IF(SFI)    120,114,114
   114  IF(SFIP1) 160,160,140

C       (MOVE COUNTERCLOCKWISE)
   120  IF(I.GT.MINI) GO TO 125
        DSIGMA=-SFI
        SFI     = 0.
        GO TO 230
   125  I       = I-3
        F       = BITS
        SFIP1   = SFI
        CALL BARC(I)
        SFI     = SFIP1+SINTVL
        GO TO 100

C       (MOVE CLOCKWISE)
   140  IF(I.LT.MAXI) GO TO 145
        DSIGMA= -SFIP1
        SFI     = SINTVL
        GO TO 230
   145  I       = I+3
        F       = BITS
        SFI     = SFIP1
        CALL BARC(I)
        SFIP1   = SFI-SINTVL
        GO TO 100

C     CALCULATE  COORDINATES OF THE NEW POINT (PROPER INTERVAL FOUND)
   160  IF(F.EQ.BITS) GO TO 230
        IF(DSI) 210,220,220
   210  F       = F*SFI/SID
        GO TO 250
   220  F       = ((SFI-SID)+(SINTVL-SFI)*F)/(SINTVL-SID)
        GO TO 250
C       (NEW INTERVAL)
   230  F       = SFI/SINTVL

   250  G       = 1.-F
        RZONLY= .FALSE.
```

120

/

```
*DECK ADPTSL
      SUBROUTINE ADPTSL(M1,MU1,MD1,J1,NEWSL)
*ADPTSL         ADD A POINT ON THE NEW STREAMLINE              -ADPTSL-
       LOGICAL                              NEWSL

C     INPUT-
C      M1     = FIELD INDEX OF THE NEW POINT
C      MU1    = UPSTREAM-M FOR NEW POINT
C      MD1    = DOWNSTREAM-M FOR NEW POINT
C      J1     = INDEX OF SL OF THE NEW POINT
C      NEWSL  = T IF A NEW SL, =F OTHERWISE

C     ACTION-
C      IF(NEWSL=T) RELOCATE FOR NEW STREAMLINE IN SL-TABLES
C      RELOCATE FOR NEW POINT IN FIELD TABLES AND CORRECT POINTERS IN JMS

       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
        DIMENSION     LIMITS(24)
        EQUIVALENCE   (LIMITS,LHO)
       COMMON /SLTAB / W(128),X2(128),SLCHN(128)
        INTEGER SLCHN
       COMMON /CIDEX / M,J,MU,MD,ISTAG


C     ADJUST STREAMLINE TABLE
       JSAV  = 999999
       IF(.NOT.NEWSL) GO TO 100
       J     = J1
       NMOVE = J-NJ-1
       CALL MOVE(3,W(J),W(J+1),NMOVE,D,
      1          X2(J),X2(J+1),NMOVE,D,
      2          SLCHN(J),SLCHN(J+1),NMOVE,0)
       NJ    = NJ+1
       JSAV  = J

C     RELOCATE FIELD POINTS AND CORRECT JMS-CHAIN
  100 CALL ADDFPT(M1,1,JSAV)

C     INSERT POINTERS IN THE JMS-TABLE
       M     = M1
       MU    = MU1
       MD    = MD1
       J     = J1
       ISTAG = 0
       CALL SAVIX

C     CORRECT UPSTREAM TO DOWNSTREAM POINTER
       M     = MU
       IF(M) 120,900,120
  120 CALL GETIX
       MD    = M1
       CALL SAVIX
  900 RETURN
       END
```

```
*DECK BARCS
      FUNCTION BARCS(NAME,IV1,IV2)
*BARCS-          ACR DISTANCE BETWEEN BOUNDARY PTS              -BARCS-

C     INPUT-
C      NAME   = BOUNDARY NAME
C      IV1,IV2=INDEX OF POINTS IN THE GIVEN BOUNDARY

C     BOUNDARY TABLE
C      INDEX- LB=LBDO,LBDE
C      LBNEXT= INCREMENT TO NEXT BOUNDARY
C      LBZ1   = INCREMENT TO THE FIRST BOUNDARY POINT (=0 BEFORE COALLATIO
C      CHNAME= CHANNEL WITH WHICH THE BOUNDARY DATA IS ASSOCIATED
C      UP     = T OR F FOR UPPER OR LOWER BOUNDARY
C      LEDEX = RELATIVE INDEX OF L.E. POINT WHEN LOWER AND UPPER SURFACE
C               CONTOURS ARE CONNECTED
C      BDNAME,LBA,LBB=NAME AND INDEX LIMITS OF SPECIFIC BOUNDARY
C                     DATA WHEN BOUNDARIES ARE COALLATED
       COMMON /CHDATA/ BDT(1),LBNEXT(1),LBZ1(1),
      1                CHNAME(1),UP(1),LEDEX(1),
      2                ZBT(1),RBT(1),ANGBT(42)
        LOGICAL        UP
        INTEGER BDT,CHNAME,BDNAME
        DIMENSION      BDNAME(1),LBA(1),LBB(1)
        EQUIVALENCE    (BDNAME,ZBT), (LBA,RBT), (LBB,ANGBT)
       COMMON /CBEAM2/ DR,DZ,YPA,YPB,F,G, DX,YQDX,ZM,RM,ANGM,CURVM,SIM,
      1                RZONLY, ANGCHD,SINTVL, YPASQ,YPAB,YPBSQ
        LOGICAL        RZONLY
C      INDEX- M=MO,NM
       COMMON /CZ     / Z(300)
       COMMON /CR     / R(300)
       COMMON /CS2    / S2(300)
       COMMON /CS1    / S1(300)
       COMMON /CPHI1  / PHI1(300)
       COMMON /CM     / JMS(300)
       COMMON /CCURV  / CURV(300)
       COMMON /CB     / B(300)
       COMMON /CIDEX  / M,J,MU,MD,ISTAG

C     INDEX IN /BDYTAB/
       LB     = LBF(NAME)

C     SUM THE ARC DISTANCES FOR INTERVALS IV1 TO (IV2-1)
       I      = LB+LBZ1(LB)+3*(IV1-1)
       IF(ISTAG.EQ.1) I=I+3
       ISTOP  = I+3*(IV2-IV1)
       S      = 0.
    75 IF(I-ISTOP)80,90,90
    80 CALL BARC(I)
       S      = S+SINTVL
       I      = I+3
       GO TO 75
    90 BARCS  = S

       RETURN
       END
```

```
      IF(NM-M) 180,130,130

  180 RETURN
      END
```

```
*DECK ADJSL
      SUBROUTINE ADJSL
*ADJSL-          ADJUST STREAMLINES BY  DS2                          -ADJSL-

C     INPUT-
C      Z,R    = COORDINATES ALONG THE STREAMLINE
C      PHI1   = STREAMLINE ANGLES
C      DS2    = DESIRED POINT MOVEMENT IN THE NORMAL DIRECTION

C     OUTPUT-
C      Z,R    = ADJUSTED COORDINATES

      COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,.
     *                LO,LESTA,  LDUM(8),
     *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
     *                LEO,LEE, LRO,LRE,LRD
       DIMENSION      LIMITS(24)
       EQUIVALENCE    (LIMITS,LHO)
      COMMON /CBITS / BITS,BLANK
      COMMON /CDS2  / DS2(300)
      COMMON /CINNER/ INRCTR,RDUM,NINNER(16),CNVF(16)
      COMMON /CMAXIT/ MAXIT,MAJCTR,GREFIN,EDUM
      LOGICAL                      GREFIN
      COMMON /CPHI1 / PHI1(300)
      COMMON /CR    / R(300)
      COMMON /CZ    / Z(300)

      CNF = CNVF(MAJCTR)
      M   = 1
 1050 R(M) = R(M) + DS2(M)*COS(PHI1(M))*CNF
      Z(M) = Z(M) - DS2(M)*SIN(PHI1(M))*CNF
      M   = M+1
      IF(M.LE.NM) GO TO 1050
      RETURN
      END
```

```
  150 WRITE (6,1150) (J,X2(J),SLCHN(J),W(J),J=1,NJ)
      CALL TABPRT(5HERASE,ERASEC,800,5)

      CALL JMSPRT
      CALL TABPRT(2HS1,S1,NM,10)
      CALL TABPRT(2HS2,S2,NM,10)
      CALL TABPRT(1HZ,Z,NM,10)
      CALL TABPRT(1HR,R,NM,10)
      CALL TABPRT(4HPHI1,PHI1,NM,10)
      CALL TABPRT(4HCURV,CURV,NM,10)
      CALL TABPRT(2HVM,VM,NM,10)
      CALL TABPRT(1HB,B,NM,10)
      CALL TABPRT(6HERASE2,AREA,1536,8)

      LSTOP = 5
      GO TO (999,999) , LSTOP
  999 RETURN
 1150 FORMAT(///1X17HSTREAMLINE TABLE-/17X32HJ       X2            SLCHN
     *    W/(I18,F12.6,6X,A6,F12.6,),)
      END
```

```
*DECK ADDFPT
      SUBROUTINE ADDFPT(INS,NPTS,JSAV1)
*ADDFPT          ADD FIELD POINTS                                  -ADDFPT-

C     INPUT-
C      INS    = FIELD INDEX OF FIRST POINT TO BE RELOCATED, INDEX OF
C               FIRST NEW POINT
C      NPTS   = NUMBER OF POINTS TO BE INSERTED
C      JSAV1  = INDEX VALUE OF NEW SL ABOVE WHICH THE FIELD J-REFERENCES A
C               TO BE INCREMENTED BY ONE, =999999 IF NO CHANGE IS TO BE MA

       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
        DIMENSION     LIMITS(24)
        EQUIVALENCE   (LIMITS,LHO)

       COMMON /CB     / B(300)
       COMMON /CM     / JMS(300)
       COMMON /CPHI1  / PHI1(300)
       COMMON /CR     / R(300)
       COMMON /CS1    / S1(300)
       COMMON /CS2    / S2(300)
       COMMON /CVM    / VM(300)
       COMMON /CZ     / Z(300)
       COMMON /CIDEX  / M,J,MU,MD,ISTAG

       M      = INS
       NPT    = NPTS
       JSAV   = JSAV1

C     RELOCATE FIELD POINTS
       NMOVE  = M-1-NM
       MTO    = M+NPT
       CALL MOVE(3,Z(M),Z(MTO),NMOVE,D,
      1            R(M),R(MTO),NMOVE,D,
      2            B(M),B(MTO),NMOVE,D)
       CALL MOVE(3,S2(M),S2(MTO),NMOVE,D,
      3            S1(M),S1(MTO),NMOVE,D,
      4            VM(M),VM(MTO),NMOVE,D)
       CALL MOVE(1,JMS(M),JMS(MTO),NMOVE,D)
C      RELOCATE FLOW ANGLES WHEN INSERTING NEW ORTHOGONAL LINES
       IF(JSAV.EQ.999999) CALL MOVE(1, PHI1(M),PHI1(MTO),NMOVE,D)
       NM     = NM+NPT

C     CORRECT THE JMS-CHAIN
       MSAV   = M
       M      = 1
  130  CALL GETIX
       IF(MU-MSAV) 140,135,135
  135  MU     = MU+NPT
  140  IF(MD-MSAV) 150,145,145
  145  MD     = MD+NPT
  150  IF(J-JSAV) 160,155,155
  155  J      = J+1
  160  CALL SAVIX
       M      = M+1
```

```
*DECK ERRORY
      SUBROUTINE ERROR1
CEDUMPX         EDUMP FOR STC EXECUTE SECTION                  -EDUMPX-

C$              ALLCOM
      COMMON /ALLCOM/ MACHA,PSA,TSA,PTA,TTA, AXIA,RGA,GAMA,
     1                MACHC,PSC,TSC,PTC,TTC, AXIC,RGC,GAMC,
     2                DAXIT,SCALEA,TTE,CHOTST
        REAL          MACHA(1),MACHC
        LOGICAL       AXIA,AXIC
        LOGICAL       CHOTST
      COMMON /CFB   / L,MA,MB,PLB,PUB,WF,CHOKE,SUBSON, NK,PLBC,PUBC,
     1                XCHOKE, TAREA,VMBC, WRQST,WCALC, QV(8),QVP(8),
     *                JSUM,VMLBSQ
        LOGICAL                              CHOKE,SUBSON
      COMMON /ERASE / ERASEC(800)
      COMMON /ERASE2/ AREA(96),AREAO(96),DISP(96),PT(96),LAMBDA(96),
     1                RHO(96),SQRTVV(96),TS(96),TT(96),VMSQ(96),
     2                VVKQKP(96),
     2                WQA(96),WSTA(96), RG(96),C2CP(96),FGR(96)
        REAL          LAMBDA
        DIMENSION     ES2(96),SDNQRM(96)
        EQUIVALENCE   (ES2,VVKQKP),(SDNQRM,RHO)
        DIMENSION     RCU(96)
        EQUIVALENCE   (RCU,LAMBDA)
C     FIELD TABLES
C      INDEX- M=MO,NM
      COMMON /CZ    / Z(300)
      COMMON /CR    / R(300)
      COMMON /CS2   / S2(300)
      COMMON /CS1 . / S1(300)
      COMMON /CPHI1 / PHI1(300)
      COMMON /CM    / JMS(300)
      COMMON /CCURV / CURV(300)

      COMMON /CB    / B(300)
      COMMON /CIDEX / M,J,MU,MD,ISTAG
C     TABLE OF INDEX LIMITS
      COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
     *                LO,LESTA,  LDUM(8),
     *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
     *                LEO,LEE, LRO,LRE,LRD
        DIMENSION     LIMITS(24)
        EQUIVALENCE   (LIMITS,LHO)
      COMMON /CVM   / VM(300)

C     STREAMLINE TABLE
      COMMON /SLTAB / W(128),X2(128),SLCHN(128)
        INTEGER SLCHN
C     BOUNDARY TABLE
C      INDEX- LB=LBDO,LBDE
C      LBNEXT= INCREMENT TO NEXT BOUNDARY
C      LBZ1  = INCREMENT TO THE FIRST BOUNDARY POINT (=0 BEFORE COALLATIO
C      CHNAME= CHANNEL WITH WHICH THE BOUNDARY DATA IS ASSOCIATED
C      UP    = T OR F FOR UPPER OR LOWER BOUNDARY
C      LEDEX = RELATIVE INDEX OF L.E. POINT WHEN LOWER AND UPPER SURFACE
C              CONTOURS ARE CONNECTED
C      BDNAME,LBA,LBB=NAME AND INDEX LIMITS OF SPECIFIC BOUNDARY
```

127

```
C                       DATA WHEN BOUNDARIES ARE COALLATED
      DIMENSION         BDT(1),LBNEXT(1),LBZ1(1),
     1                  CHNAME(1),UP(1),LEDEX(1),
     2                  ZBT(1),RBT(1),ANGBT(42)
        LOGICAL         UP
        INTEGER BDT,CHNAME,BDNAME
        DIMENSION       BDNAME(1),LBA(1),LBB(1)
        EQUIVALENCE     (BDNAME,ZBT), (LBA,RBT), (LBB,ANGBT)
C     FLOW ADJUSTMENT TABLE
C      INDEX- LF=LFO,LFE
C      NFCOLS= 8
C      X1F    = ORTHOGONAL COORDINATE
C      X2F    = STREAMLINE COORDINATE OF SL EMINATING FROM T.E.
C      X1BF   = X1-COORDINATE OF CHOKE STATION OF FLOW BELOW T.E.
C      X1AF   = X1-COORDINATE OF CHOKE STATION OF FLOW ABOVE T.E.
C      S1F    = S1-COORDINATE OF T.E. (UPPER SURFACE). THIS ITEM
C               IS USED WHEN INTERPOLATING FOR WAKE DELTA-STAR.
C      LFB,LFA=INDICES OF STATIONS BELOW AND ABOVE T.E.
C      NCHB,NCHA=NUMBER OF CHANNELS BELOW AND ABOVE T.E.
C      LRF    = INDEX OF DUMMY ORTCHN LIST FOR THE T.E.
C      LRXF   = INDEX OF LAST CHANNEL BELOW THE T.E.
C      JORDER= 0 IF TOTAL FLOW AT X1F IS GIVEN
C            = 2 IF FLOW ABOVE T.E. IS GIVEN
C            = 1 IF FLOW BELOW T.E. IS GIVEN
C      JORDER= -1 IF FLOW AT X1F IS CHOKED AND SINGLE CHANNEL
      DIMENSION         X1F(1),X2F(1),X1BF(1),X1AF(1),
     1                  S1F(1),NCHB(1),NCHA(1),JORDER(1),VNR(12)
        EQUIVALENCE     (LFB,X1BF),(LFA,X1AF),(LRF,NCHB),(LRXF,NCHA)
        DIMENSION       LFB(1),LFA(1),LRF(1),LRXF(1)
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
      COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
     1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
     1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
     3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
        LOGICAL         PRIM
        DIMENSION       SCHOKE(1)
        EQUIVALENCE     (SCHOKE,DWDV)

        EQUIVALENCE    (BDT,X1F,X1), (LBNEXT,X2F,LNEXT), (LBZ1,X1BF,MLB)
        EQUIVALENCE    (CHNAME,X1AF,MUB), (UP,S1F,PRIM)
        EQUIVALENCE    (LEDEX,NCHB,TYPELB), (ZBT,NCHA,NAMELB)
        EQUIVALENCE    (RBT,JORDER,ILB), (ANGBT,VNR,FLB)

      COMMON /CTABPR/ I1TAB

      CALL TABPRT(6HALLCOM,MACHA,20,8)
      CALL TABPRT(3HCFB,L,33,4)
      CALL TABPRT(5HCIDEX,M,5,5)
      I1TAB = LBDO
      CALL TABPRT(6HBDYTAB,BDT,LBDE,3)
      I1TAB = LFO
      CALL TABPRT(6HCADJWF,X1F,LFE,8)
      I1TAB = LO
      CALL TABPRT(6HSTATAB,X1,LESTA,5)
```

```
      CALL LSPFIT(AREA,FPY,NK, AREA,SPY,NK, -1)
      DO 630 K=1,NK
      STX(K)= SVX(K)+SPX(K)
  630 STY(K)= SVY(K)+SPY(K)

      KA     = 1
      DO 640 LL=1,LQ
      J      = J1(LL+1)-1
      K      = K1(LL+1)-1
      IF(MU.NE.0) GO TO 635
      STXU(J)=STX(K)-STX(KA)
      STYU(J)=STY(K)-STY(KA)
  635 IF(MD.NE.0) GO TO 640
      STYD(J)=STY(K)-STY(KA)
      STXD(J)=STX(K)-STX(KA)
  640 KA     = K

      IF(PRPRN.EQ.(-1)) GO TO 800
      WRITE (6,1700) SVX(NK),SVY(NK),SPX(NK),SPY(NK),STX(NK),STY(NK)
      LINES = LINES+4
 1700 FORMAT(/6X25HSUM-VM*COS(PHI)*DFLOW    =F10.2,36X,25HSUM-VM*SIN(PHI)
     **DFLOW    =F10.2,/6X25HSUM-(P-PSO)*COS(PHI)*DA =F10.2,36X,25HSUM-(P
     *-PSO)*SIN(PHI)*DA =F10.2,/6X25HTOT AXIAL MOMENTUM FLUX =F10.2,36X,
     *25HTOTAL Y-MOMENTUM FLUX    =F10.2,)

C     RELOCATE DATA INTO THE M-ARRAYS
  800 CALL MOVE(2, MACH,MACHM(MA),NK,1, PS,PSM(MA),NK,1)
      CALL MOVE(2,PT,PTM(MA),NK,1, TT,TTM(MA),NK,1)

C     FILL IN STAGNATION POINT VALUES
      IF(MLB(L).EQ.MA) GO TO 820
      M      = MLB(L)
      CALL GETIX
      MACHM(M)=0.
      PTM(M)=PTM(MU)
      PSM(M)=PTM(M)
      TTM(M)=TTM(MU)
      VMF(M)= 0.
  820 IF(MUB(L).EQ.MB) GO TO 830
      M      = MUB(L)
      CALL GETIX
      MACHM(M)=0.
      PTM(M)=PTM(MU)
      PSM(M)=PTM(M)
      TTM(M)=TTM(MU)
      VMF(M)= 0.

C     INDEX TO NEXT STATION
  830 L      = L+LNEXT(L)
      IF(L.LT.LESTA) GO TO 500

      RETURN
      END
      OVERLAY(STC,3,0)
```

```
*DECK STCXX
      PROGRAM STCXX
      COMMON /CMAXIT/ MAXIT,MAJCTR,GREFIN,EDUM
      COMMON /SELECT/ LENTRY
    1 GO TO  (5,10,15,20,17) , LENTRY
C     INITIAL DISTANCE ALONG SL-S
    5 CALL SLC
      GO TO 25
C     REFINE GRID
   10 CALL REFINE
      GO TO 25
C     SLC,PTMOVE,ADJWF
   15 CALL SLC
   17 CALL PTMOVE
      CALL SPC
      CALL FARFLD
      GO TO 25
C     ADJUST SL POSITION, CALC. FAR FIELD VELOCITIES
   20 CALL ADJSL
   25 RETURN
      END
```

```
      IFIELD= 0
      JSUM  = 0
      LINES = 64
      LINEA = 0
      L     = LO
  500 PLB   = 0.
      PUB   = 0.
      WF    = 0.

C     SUBSONIC/SUPERSONIC BRANCH SELECTION
      M     = MLB(L)
      CALL GETIX
      JA    = J
      MAA   = M
      M     = MUB(L)
      CALL GETIX
      JB    = J
      MBB   = M
      IF(JSUM.EQ.0) SUBSON=.TRUE.
      IF(SSEF) SUBSON=.FALSE.
      IF(SCHOKE(L).NE.XCHOKE) GO TO 510
      IF(SSDF) SUBSON=.FALSE.
      JSUM  = JA+256*JB

C     EXECUTE FLOW BALANCE
  510 CALL FLOBAL
      IF(TYPELB(L).EQ.TE .OR. TYPEUB(L).EQ.TE) JSUM=0

C      BRANCH AND ASTERP ARE PRINTOUT INDICATORS
      DATA DBSTAR/2H**/, SUB/3HSUB/, SUPER/5HSUPER/, ICHOKE/5HCHOKE/
  501 ASTERP= BLANK
      IF(PRIM(L)) ASTERP=DBSTAR
      BRANCH= SUPER
      IF(SUBSON) BRANCH=SUB
      IF(SCHOKE(L).EQ.XCHOKE) BRANCH=ICHOKE

      CALL SETM(1,BLANK, CHANLS,10)
      CALL MOVE(2,ZF(MA),Z,NK,1, RF(MA),R,NK,1)
      CALL MOVE(2,CURVF(MA),CURV,NK,1, VMF(MA),VM,NK,1)

      LQ    = 0
      K     = 1
      M     = MA
  520 FLOW(K)=WSTA(K)*CG
      PHI(K)= PHII(M)*TODEG
      QGAM  = FGR(K)/(1.+FGR(K))
      MACH(K)=VM(K)*SQRT(QGAM/(RG(K)*TS(K)))
      AQAREF(K) = R(K)
      IF ( AXIA ) AQAREF(K) = PI*R(K)*R(K)
      PS(K) = RHO(K)*RG(K)*TS(K)
      PSQPO(K)=PS(K)/PSA
      PSQPT(K)=PS(K)/PT(K)
      TSQTT(K)=TS(K)/TT(K)
C     CP MUST FOLLOW USE OF RG
      CP(K)= (PS(K)-PSA)*QO
      CALL GETIX
      XI2(K)= X2(J)
      IF(SLCHN(J).EQ.CHANLS(LQ)) GO TO 530
```

```
      LQ     = LQ+1
      J1(LQ)= J
      K1(LQ)= K
      CHANLS(LQ)=SLCHN(J)
      IF(LQ.GT.1) FLOWMX(LQ-1)=FLOW(K)
      I      = 0
  525 I      = I+1
      IF(SLCHN(J).NE.ICHN(I).AND.I.LT.IC) GO TO 525
      QPTO   = 1./WPTO(I)
  530 PTQPTO(K)=PT(K)*QPTO
      K      = K+1
      M      = M+1
      IF(K.LE.NK) GO TO 520
      J1(LQ+1)=J+1
      K1(LQ+1)=K
      FLOWMX(LQ)=FLOW(K-1)
      LQS    = 0
  533 LQS    = LQS+1
      KB     = K1(LQS)
      KE     = K1(LQS+1)-1
      FLMX   = 1./FLOWMX(LQS)
      DO 535 K=KB,KE
  535 PFLOW(K)=FLOW(K)*FLMX
      IF(LQS.LT.LQ) GO TO 533

      XI1    = X1(L)
      IF(PRPRN.EQ.(-1)) GO TO 610
      CALL FHEAD(LINEA+NK)
      LINEA = 4
      IF(.NOT.PRIM(L)) LINEA=8
      WRITE (6,1600) XI1,ASTERP,CHANLS,BRANCH,
     1 (XI2(K),PFLOW(K),Z(K),R(K),PHI(K),CURV(K),PSQPO(K),PSQPT(K),
     2 TSQTT(K),CP(K),MACH(K),AQAREF(K),PTQPTO(K),K=1,NK)

 1600 FORMAT (/25H STATION COORDINATE, XI1=,F7.3,A2,13H  CHANNELS-  ,
     110(A6,2X),A5//   5X,13HXI2 STRM FNCT,6X,3HX,Z,8X,3HY,R,8X,3HPHI,
     16X,4HCURV,6X,21HPS/PO   PS/PT   TS/TT,6X,2HCP,6X,4HMACH,6X,
     3 6H  AREA,3X,6HPT/PTO   / (2X,F6.3,F10.3,F12.5,F11.5,F9.3,F11.5,
     4 F9.3,2F8.3,F10.3,F9.4,F11.3,F9.3,7X,),)


  610 IF(.NOT.PRIM(L)) GO TO 800
      M      = MA
      DO 620 K=1,NK
      COSPHI= COS(PHI1(M))
      SINPHI= SIN(PHI1(M))
      FVX(K)=VM(K)*COSPHI
      FVY(K)=VM(K)*SINPHI
      FPX(K)=(PS(K)-PSA)*COSPHI
      FPY(K)=(PS(K)-PSA)*SINPHI
  620 M      = M+1
      SVX(1)= 0.
      SVY(1)= 0.
      SPX(1)= 0.
      SPY(1)= 0.
      CALL LSPFIT(WSTA,FVX,NK, WSTA,SVX,NK, -1)
      CALL LSPFIT(WSTA,FVY,NK, WSTA,SVY,NK, -1)
      CALL LSPFIT(AREA,FPX,NK, AREA,SPX,NK, -1)
```

```
*DECK WRIOUT
      SUBROUTINE WRIOUT
*WRIOUT         WRITE STC OUTPUT DATA                              -WRIOUT-

       COMMON /ALLCOM/ MACHA,PSA,TSA,PTA,TTA, AXIA,RGA,GAMA,
      1                MACHC,PSC,TSC,PTC,TTC, AXIC,RGC,GAMC,
      2                DAXIT,SCALEA,TTE,CHOTST
        REAL           MACHA(1),MACHC
        LOGICAL        AXIA,AXIC
        LOGICAL        CHOTST
       COMMON /CFB    / L,MA,MB,PLB,PUB,WF,CHOKE,SUBSON, NK,PLBC,PUBC,
      1                XCHOKE, TAREA,VMBC, WRQST,WCALC, QV(8),QVP(8),
      *                JSUM,VMLBSQ
        LOGICAL                          CHOKE,SUBSON
       COMMON /CSS    / SSFML,SSEF,SSEANG,SSDF,SSFEND,SSFND1
      1                ,SSDLE,A4FACT,BRLX,CURRLX
        INTEGER        SSFML
        LOGICAL              SSEF,       SSDF,   SSDLE
C       SSFML = SUPERSONIC CURVATURE FORMULA NUMBER
C       SSEF  = SUPERSONIC ENTERING FLOW, T OR F
C       SSEANG= ENTERING FLOW ANGLE (DEGREES) FOR SSEF=T
C       SSDF  = SUPERSONIC DISCHARGE FLOW, T OR F
C       SSFEND= SUPERSONIC BEAM DOWNSTREAM END CONDITION, =0,1 FOR PARABOL
C       SSFND1= SUPERSONIC BEAM UPSTREAM END CONDITION, =0,1, FOR PARABOLA
C       SSDLE = SS FLOW BELOW AND AFT OF LE PT, T OR F
C       A4FACT= CENTRAL POINT INFLUENCE COEFFICIENT FACTOR
C       BRLX  = B-RELAXATION FACTOR
C       CURRLX= CURVATURE RELAXATION FACTOR
       COMMON /ERASE2/ AREA(96),AREAO(96),DISP(96),PT(96),LAMBDA(96),
      1                RHO(96),SQRTVV(96),TS(96),TT(96),VMSQ(96),
      2                VVKQKP(96),
      2                WQA(96),WSTA(96), RG(96),C2CP(96),FGR(96)
        REAL           LAMBDA
       COMMON /ERASE3/ J1(10),K1(10),CHANLS(10),PS(96),MACH(96),FLOW(96)
        DIMENSION      XI2(96),Z(96),R(96),PHI(96),CURV(96),PSQPO(96),
      *                VM(96),FVX(96),FVY(96),FPX(96),FPY(96),SVX(96),
      *                SVY(96),SPX(96),SPY(96),STX(96),STY(96)
        EQUIVALENCE    (AREAO,XI2,FVX,STX), (DISP,Z,FVY,STY),
      *                (SQRTVV,R,FPX), (VMSQ,PHI,FPY), (VVKQKP,CURV,SVX),
      *                (WQA,PSQPO,SVY), (C2CP,VM,SPX), (FLOW,SPY)
        REAL           MACH
        DIMENSION      X(1),Y(1)
        EQUIVALENCE    (X,Z),(Y,R)
C   NEW VARIABLES FOR NASA VERSION ONLY
C   CAN USE FGR IF NEEDED
         DIMENSION     PFLOW(96),PSQPT(96),TSQTT(96),CP(96),AQAREF(96),
      *                PTQPTO(96),FLOWMX(10)
         EQUIVALENCE   (FLOW,PFLOW), (LAMBDA,PSQPT), (TS,TSQTT),
      *                (RHO,CP) ,(FGR,AQAREF), (RG,PTQPTO)
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
        DIMENSION      LIMITS(24)
        EQUIVALENCE    (LIMITS,LHO)
       COMMON /CBEND / NBCB(2),ANGE(2),CURVE(2),FB(2)
       COMMON /CBITS / BITS,BLANK
```

```
      COMMON /CCUBE  / NBC(2),C1(2),C2(2),FEND(2)
      COMMON /CGRAV  / CG
      COMMON /CPI    / PI,TWOPI,PIO2,PIO4,TODEG,TORAD
      COMMON /CREFIN/ SG1,SG2,VMG1,VMG2
     1,               NGR,NGZ, SGR(10),GR(10), SGZ(10),GZ(10)
      COMMON /SLTAB / W(128),X2(128),SLCHN(128)
       INTEGER SLCHN
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
      COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
     1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
     1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
     3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
       LOGICAL         PRIM
       INTEGER TYPELB,TYPEUB
       DIMENSION       SCHOKE(1)
       EQUIVALENCE     (SCHOKE,DWDV)

      COMMON /BCOMMN/ PROGM(9),FILIN,FILOT
       LOGICAL                FILIN,FILOT
      COMMON /ADAM01/ NAME(6),ADDRES(6),TITLE(6),IDENT(6)
      COMMON /CCURV  / CURVF(300)
      COMMON /CDS2   / MACHM(300)
       REAL            MACHM
      COMMON /CPHI1  / PHI1(300)
      COMMON /CPRINT/ PRTES2,PRTB,PRTA,PREFIN,PREFN2,SSONIC,PDUM(10)
      COMMON /CPRPRN/ PRPRN
       INTEGER         PRPRN
      COMMON /CB     / PSM(300)
      COMMON /CS2    / PTM(300)
      COMMON /CR     / RF(300)
      COMMON /CRHS   / TTM(300)
      COMMON /CVM    / VMF(300)
      COMMON /CZ     / ZF(300)

      COMMON /CIDEX  / M,J,MU,MD,ISTAG
      COMMON /CLINES/ LINES,OMITFK,PTITLE(6)

      COMMON /CFRFLD/ FSAV(300), STXU(128),STXD(128),STYU(128),STYD(128)
      COMMON /CHNFPT/ ICHN(10),WTFS(10),WTFA(10),WPTO(10),WTTO(10), IC

       INTEGER DBSTAR,SUB,SUPER,BLANK,BRANCH,CHANLS,ASTERP,TE
      LOGICAL         UPSTRM,DNSTRM
      DATA TE/2HTE/

      PIINV = 1./PI
      QO    = 0.
      IF(MACHA.LE..1) GO TO 95
      IF(GAMA.NE.0.) GO TO 92
      QO    = (RGA*TSA)/(PSA*MACHA*MACHA)
      GO TO 95
   92 QO    = 2./(GAMA*PSA*MACHA*MACHA)

C     BEGIN LOOP THROUGH STATIONS
   95 CHOKE = .FALSE.
```

```
  160 M      = MD
      IF(M.GT.0) GO TO 124
      GO TO 180

C     APPROACH STREAMLINE
  170 XKEYB    =ASL
      GO TO 200
C     BODY SURFACE
  180 XKEYB    =XK5SV
      GO TO 200
C     TRAILING STREAMLINE
  190 XKEYB    =XK5SV
      XK5SV = TSL

  200 IF(XKEYB    .EQ.TSL) GO TO 220
      IF(.NOT.LOWER) GO TO 220
      LB     = LBF(NAMELB(L))
      IF(LEDEX(LB).EQ.0) GO TO 220
C      LOOP TO FIND BOUNDARY NAME OF UPPER SIDE OF L.E.
      LBX    = LB
  214 IF(LBA(LBX).GE.LEDEX(LB)) GO TO 220
      LBX    = LBX+3
      IF(LBX.LT.(LB+LBZ1(LB))) GO TO 214
      CALL ERROR1
  220 SPDA(1)=SPDASV
      CALL LSUM(AW,PSMPO,NI, SPDA)
      SPDASV= SPDA(NI)
      ARM      = RM
      IF ( AXIA ) ARM = PI*RM*RM
      DO 225 I=1,NI
      AW(I) = (AW(I)-ARM)/ARM
  225 CDPI(I) = SPDA(I) *QO/ARM
      ADDG   = SPDASV*QO/ARM
  230 LINES = 64
      CALL FHEAD(NI+6)
      KUP    = 2
      IF(LOWER) KUP=1
      CHN    = SLCHN(J2)
      XI2    = X2(J2)
      SWORG = 0.
      WRITE (6,1200) LOWUP(KUP),CHN,XI2, (XI1(I),SW(I),ZW(I),RW(I),
     * ANGW(I),CURVW(I),PSQPO(I),CP(I),PSQPT(I),MACH(I),CDPI(I),AW(I),
     * PTQPTO(I),I=1,NI)
 1200 FORMAT (/2X,A6,17H BOUNDARY TO CHN=,A6,31H,   STREAMLINE COORDINAT
     *E, XI2=,F7.3,1H.//   5X,3HXI1,6X,3HSIW,7X,5HXW,ZW,6X,5HYW,RW,5X,
     * 4HANGW,5X,5HCURVW,5X,5HPS/PO,5X,2HCP,4X,5HPS/PT,4X,4HMACH,5X,
     * 4HCDPI,14H (A-AMAX)/AMAX,8H  PT/PTO  / (2X,2F8.3,F12.5,F11.5,
     * F8.3,F11.5,2F9.3,F7.3,2F9.4,F14.3,F8.3,),)

      WRITE (6,1210) TTQTTO
 1210 FORMAT (/6X,8HTT/TTO =,F9.3)
      IF ( XKEYB.EQ.ASL ) WRITE (6,1220) ADDG
 1220 FORMAT (/6X,15HADDITIVE DRAG =,F9.4)

      IF(MD.GT.0) GO TO 123

C     INTEGRAL MOMENTUM BALANCE ON THE CHANNEL
      IF(.NOT. LOWER) GO TO 310
```

```
      FLB    = SPDASV
      GO TO 110
  310 FUB    = SPDASV
      FTOT   = STXU(J2)+FLB+FUB
      FERR   = FTOT-STXD(J2)
      WRITE (6,1300) CHN,STXU(J2),FLB,FUB,FTOT,STXD(J2),FERR
 1300 FORMAT(/1X32HINTEGRAL MOMENTUM BALANCE,  CHN=A6,2X19H(AXIAL FORCES
     * ONLY)/6X31HENTERING MOMENTUM                =F11.4,/6X31HLOWER BOUND
     *ARY PRESSURE FORCE =F11.4,/6X31HUPPER BOUNDARY PRESSURE FORCE =F11
     *.4,/12X12HSUM OF ABOVEF24.4,/6X31HLEAVING MOMENTUM                =F
     *11.4,/12X25HERROR                          =F11.4,)

      J2     = J2+1
      IF(J2.LE.NJ) GO TO 105
      RETURN
      END
```

```
      COMMON /CFB    / L,MA,MB,PLB,PUB,WF,CHOKE,SUBSON, NK,PLBC,PUBC,
     1                 XCHOKE, TAREA,VMBC, WRQST,WCALC, QV(8),QVP(8),
     *                 JSUM,VMLBSQ
       LOGICAL                              CHOKE,SUBSON
C      INDEX- M=MO,NM
       COMMON /CZ     / Z(300)
       COMMON /CR     / R(300)
       COMMON /CS1    / S1(300)
       COMMON /CPHI1  / PHI1(300)
       COMMON /CM     / JMS(300)
       COMMON /CCURV  / CURV(300)
       COMMON /CIDEX  / M,J,MU,MD,ISTAG
       COMMON /CVM    / VM(300)
       COMMON /CDS2   / MACHM(300)
       REAL             MACHM
       COMMON /CB     / PSM(300)
       COMMON /CS2    / PTM(300)
       COMMON /CRHS   / TTM(300)

       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
     *                 LO,LESTA,  LDUM(8),
     *                 MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
     *                 LEO,LEE, LRO,LRE,LRD
       DIMENSION       LIMITS(24)
       EQUIVALENCE     (LIMITS,LHO)
       COMMON /CBEND / NBCB(2),ANGE(2),CURVE(2),FB(2)
       COMMON /CBITS / BITS,BLANK
       COMMON /CCUBE / NBC(2),C1(2),C2(2),FEND(2)
       COMMON /CGRAV / CG
       COMMON /CPI    / PI,TWOPI,PIQ2,PIQ4,TODEG,TORAD
       COMMON /CREFIN/ SG1,SG2,VMG1,VMG2
      1,               NGR,NGZ, SGR(10),GR(10), SGZ(10),GZ(10)
       COMMON /SLTAB / W(128),X2(128),SLCHN(128)
       INTEGER SLCHN
       COMMON /CHNFPT/ ICHN(10),WTFS(10),WTFA(10),WPTO(10),WTTO(10), IC

       INTEGER HLE,HTE,ASL,BDY,TSL,CHNN,CHN,XK5SV,XKEYB,BLANK

       LOGICAL         DOUBLE,LOWER,UPPER
       DIMENSION       LOWUP(2)
       DATA LOWUP/5HLOWER,5HUPPER/
       DATA HLE,HTE/2HLE,2HTE/, ASL,BDY,TSL/3HASL,3HBDY,3HTSL/

C      DEFINE REFERENCE DYNAMIC PRESSURE, ETC
       QO     = 0.
       IF(MACHA.LE..1) GO TO 95
       IF(GAMA.NE.0.) GO TO 92
       QO     = (RGA*TSA)/(PSA*MACHA*MACHA)
       GO TO 95
    92 QO     = 2./(GAMA*PSA*MACHA*MACHA)

C      BEGIN LOOP THROUGH CHANNELS
    95 LINES = 64
       IUP   = 4
       NCHN  = 1
       J2    = 1
   105 CHNN  = SLCHN(J2)
       LOWER = .TRUE.
```

137

```
      I      = 0
  107 I      = I+1
      IF(CHNN.NE.ICHN(I) .AND. I.LT.IC) GO TO 107
      OPTO   = 1./WPTO(I)
      QTTO   = 1./WTTO(I)
      GO TO 122
  110 J2     = J2+1
      IF(J2.EQ.NJ .OR. SLCHN(J2+1).NE.CHNN) GO TO 120
      GO TO 110
  120 LOWER = .FALSE.

C     BUILD I-SUBSCRIPTED ARRAYS
  122 M      = MBEGIN(J2)
      L      = 0
      SPDASV= 0.
      XK5SV = BDY
  123 I      = 1
      SWORG = S1(M)
      PTO    = PTM(M)
      TTO    = TTM(M)
      TTQTTO= TTM(M)*QTTO
  124 DOUBLE= .FALSE.
  125 SW(I) = S1(M) - SWORG
      ZW(I) = Z(M)
      RW(I) = R(M)
      ANGW(I)=PHI1(M)*TODEG
      CURVW(I)= CURV(M)
      PS     = PSM(M)
      PSQPT(I)=PS/PTM(M)
      PTQPTO(I)=PTM(M)*QPTO
      MACH(I)=MACHM(M)
      AW(I) = RW(I)
      IF( AXIA ) AW(I)=PI*RW(I)*RW(I)
      PSQPO(I)=PS/PSA
      PSMPO(I)=PS-PSA
      CP(I) = PSMPO(I)*QO
      IF(LOWER) PSMPO(I)=-PSMPO(I)
      CALL GETIX
      CALL STAND(M,L,UPPER)
      XI1(I)=X1(L)
      NI     = I
      I      = I+1
      IF(NI.EQ.1) GO TO 160

C       CHECK FOR LEADING EDGE POINT
      IF(ISTAG.NE.1) GO TO 140
      IF(TYPELB(L).EQ.HLE .OR. TYPEUB(L).EQ.HLE) GO TO 170
C       ISTAG=1
      IF(DOUBLE) GO TO 160
      DOUBLE= .TRUE.
      GO TO 125

C       CHECK FOR TRAILING EDGE POINT
  140 IF(ISTAG.NE.2) GO TO 160
C       ISTAG=2
      IF(TYPELB(L).EQ.HTE .OR. TYPEUB(L).EQ.HTE) GO TO 190

C       ISTAG=0,3 OR DOUBLE=T
```

```
*DECK USECDW
      BLOCK DATA USECDW
*USECDW         REPLACE STCW USE CARDS
      COMMON /ERASE3/ WDUM(318)
      END
```

```
*DECK WRIBDY
      SUBROUTINE WRIBDY
*WRIBDY          WRITE OUTPUT FOR EACH BOUNDARY                    -WRIBDY-

      COMMON /BCOMMN/ PROGM(9),FILIN,FILOT
       LOGICAL                FILIN,FILOT
      COMMON /ADAM01/ NAME(6),ADDRES(6),TITLE(6),IDENT(6)
      COMMON /CLINES/ LINES,OMITFK,PTITLE(6)
      COMMON / CNORM / RHL,RM,AHL,ARM
      COMMON /ERASE2/ XI1(100),SW(100),ZW(100),RW(100),ANGW(100),
     *                CURVW(100),VE(100),MACH(100),PSQPO(100),CP(100),
     *        PSQPT(100),PTQPTO(100), TT(100),AW(100),SPDA(100)
C  NEW VARIABLES FOR NASA VERSION ONLY--PSQPT AND PTQPTO
      COMMON /ERASE3/ AQAN(100),CDPI(100),PSMPO(100)
       REAL           MACH
       DIMENSION      XW(1),YW(1)
       EQUIVALENCE    (XW,ZW),(YW,RW)
      COMMON /CFRFLD/ FSAV(300), STXU(128),STXD(128),STYU(128),STYD(128)

      COMMON /ALLCOM/ MACHA,PSA,TSA,PTA,TTA, AXIA,RGA,GAMA,
     1                MACHC,PSC,TSC,PTC,TTC, AXIC,RGC,GAMC,
     2                DAXIT,SCALEA,ITE,CHOTST
       REAL           MACHA(1),MACHC
       LOGICAL        AXIA,AXIC
       LOGICAL        CHOTST
C     BOUNDARY TABLE
C      INDEX- LB=LBDO,LBDE
C      LBNEXT= INCREMENT TO NEXT BOUNDARY
C      LBZ1  = INCREMENT TO THE FIRST BOUNDARY POINT (=0 BEFORE COALLATIO
C      CHNAME= CHANNEL WITH WHICH THE BOUNDARY DATA IS ASSOCIATED
C      UP    = T OR F FOR UPPER OR LOWER BOUNDARY
C      LEDEX = RELATIVE INDEX OF L.E. POINT WHEN LOWER AND UPPER SURFACE
C              CONTOURS ARE CONNECTED
C      BDNAME,LBA,LBB=NAME AND INDEX LIMITS OF SPECIFIC BOUNDARY
C                     DATA WHEN BOUNDARIES ARE COALLATED
       DIMENSION      BDT(1),LBNEXT(1),LBZ1(1),
     1                CHNAME(1),UP(1),LEDEX(1),
     2                ZBT(1),RBT(1),ANGBT(42)
       LOGICAL        UP
       INTEGER BDT,CHNAME,BDNAME
       DIMENSION      BDNAME(1),LBA(1),LBB(1)
       EQUIVALENCE    (BDNAME,ZBT), (LBA,RBT), (LBB,ANGBT)
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL   = SHARP CORNER INDICATOR (BLDTBS)
C      MCL   = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
       COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
     1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
     1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
     3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
       LOGICAL        PRIM
       INTEGER TYPELB,TYPEUB
       DIMENSION      SCHOKE(1)
       EQUIVALENCE    (SCHOKE,DWDV)
       EQUIVALENCE    (X1,BDT),(LNEXT,LBNEXT),(MLB,LBZ1),(MUB,CHNAME)
       EQUIVALENCE    (PRIM,UP),(TYPELB,LEDEX),(NAMELB,ZBT),(ILB,RBT)
       EQUIVALENCE    (FLB,ANGBT)
```

```
      LH     = LHP
      GO TO 122
  124 IF(PTO(LH).NE.BITS .AND. PTO(LH).NE.0.) WPTO(IC)=PTO(LH)
      IF(TTO(LH).NE.BITS .AND. TTO(LH).NE.0.) WTTO(IC)=TTO(LH)
  128 IF(J2.LT.NJ) GO TO 100
  130 WRITE (6,1130) (ICHN(I),WTFS(I),WTFA(I),WPTO(I),WTTO(I),I=1,IC)
 1130 FORMAT (/49H CHANNEL FLOW RATES, PRESSURES, AND TEMPERATURES-//
     * 16X,9HSPECIFIED,5X,8HADJUSTED,7X,6HPT/PSO,7X,6HTT/TSO /
     * (6X,A6,4F13.4,),)

      IF(.NOT.FILOT) RETURN
      REWIND  NTAPN
      WRITE (NTAPN) STCFIL,(LIMITS(I),I=1,24)
      WRITE (NTAPN) ((IDENT(I),I=1,6),AXI,RG,GAM,MACHO,PSO,TSO,PTO,TTO,
     1  PRPKN,TTE,CHOTST,MAXIT,MAJCTR, (NINNER(I),I=1,16), VELPOT,ICOB,
     2  NODENS,RN,NGR,NGZ,(SGR(I),I=1,40),VMG1,VMG2, INRCTR, SLS,SG21,
     3  NBCIN(1),NBCIN(2),ACF(1),ACF(2), SSFML,SSEF,SSEANG,SSDF,SSFEND,
     4  SSFND1,SSDLE,A4FACT,BRLX,CURRLX,TSIC,(FARFLD(I),I=1,8),
     * RHOC,RHOCSS,RHL,RM,
     5  (ZP(I),I=1,28), (TABLES(I),I=1,LESTA), (B(I),I=1,NM), (JMS(I),
     6  I=1,NM), (S1(I),I=1,NM), (S2(I),I=1,NM),(ZF(I),I=1,NM), (RF(I),
     7  I=1,NM); (VMF(I),I=1,NM), (W(I),I=1,NJ), (X2(I),I=1,NJ),
     8  (SLCHN(I),I=1,NJ),IOLRL,MAXSWP,TOLES2,TOLINR,SG1MIN,DS1DMP,
     A  DS1RMO,(CRX(I),I=1,6), RHOBAS,RHOAMP,IADM,NTHKX,NTHKY,
     B  (THKX(I),I=1,118)  )
      NTSAV = NTAPO
      NTAPO = NTAPN
      NTAPN = NTSAV
      RETURN
      END
      OVERLAY(STC,2,3)
```

```
      WRITE(6,1005) SSFML,SSFEND,SSFND1,SSEANG,SSEF,SSDF,SSDLE
 1005 FORMAT(43H CURVATURE CALCULATION FOR SUPERSONIC FLOW-/
     16X,7HSSFML =,I8,19H   (FORMULA NUMBER)/
     26X,7HSSFEND=,F8.3,43H   (DOWNSTREAM END CONDITION, SSFML=2 ONLY)/
     36X,7HSSFND1=,F8.3,41H   (UPSTREAM END CONDITION, SSFML=2 ONLY)/
     46X,7HSSEANG=,F8.3,43H   (INLET FLOW ANGLE, DEGREES, SSEF=T ONLY)/
     5/38H SUBSONIC/SUPERSONIC BRANCH SELECTION-/
     66X,7HSSEF  =,L8,37H   (SUPERSONIC ENTERING FLOW, T OR F)/
     76X,7HSSDF  =,L8,56H   (SUPERSONIC FLOW DOWNSTREAM OF CHOKE STATION
     *, T OR F)/       6X,7HSSDLE =,L8,58H   (SUPERSONIC FLOW BELOW AND
     8AFT OF A L.E. POINT, T OR F)  )

      WRITE (6,1010) (GR(I),I=1,NGR)
      WRITE (6,1011) (SGR(I),I=1,NGR)
      IF(NGZ.EQ.0) GO TO 65
      WRITE (6,1012) (GZ(I),I=1,NGZ)
      WRITE (6,1013) (SGZ(I),I=1,NGZ)
   65 WRITE (6,1014) VMG1,VMG2,CRX
 1010 FORMAT(/1X19HGRID SIZE CRITERIA-/6X7HNGR/GR=10F8.2)
 1011 FORMAT (6X,7HSGR   =,10F8.2)
 1012 FORMAT (/6X,7HNGZ/GZ=,10F8.2)
 1013 FORMAT(6X,7HSGZ   =,10F8.2)
 1014 FORMAT(/6X,7HVMG1  =,F8.2,25X,7HVMG2  =,F8.2//6X,7HCRX   =,6F8.3)

      WRITE (6,1030) NM,MAXNM, LESTA,MAXLE, NJ,MAXNJ
 1030 FORMAT(/1X19HMEMORY UTILIZATION-/24X17HUSED    AVAILABLE/6X11HGRID
     * POINTS111,I10,/6X6HTABLES116,I10,/6X11HSTREAMLINES111,I10,)

      ATLDS2= CLEN*TOLES2
      WRITE(6,1040) MAXIT,NREFIN,INRCTR,TOLINR,TOLES2,CLEN,ATLDS2,ES2MX,
     1              DS1DMP,NODENS
 1040 FORMAT (/18H CONVERGENCE DATA-/
     16X,7HMAXIT =,I8,3X,20H(MAXIMUM ITERATIONS)/
     26X,7HNREFIN=,I8,34H - NUMBER OF REFINEMENT ITERATIONS/ 6X,7HINRCTR
     3=,I8,56H - NUMBER OF ADDITIONAL ITERATIONS AFTER LAST REFINEMENT//
     46X,7HTOLINR=,  E8.1,47H   (INNER ITERATION TOLERANCE ON S.L. MOVEM
     4ENT)/ 6X,7HTOLES2=,  E8.1,37H   (FINAL TOLERANCE ON S.L. MOVEMENT)
     5/6X,7HCLEN  =,  F8.3,52H - CHARACTERISTIC LENGTH BASED ON GRID SIZ
     6E CRITERIA/   E21.1,38H - ABSOLUTE TOLERANCE ON S.L. MOVEMENT/
     76X,7HMAXES2=,  E8.1,42H - LARGEST S.L. MOVEMENT ON LAST ITERATION/
     8/6X,7HDS1DMP=,  F8.3,54H   (STREAMWISE PT MOVEMENT DAMPING, =0 FOR
     9 NO DAMPING)/ 6X,7HNODENS=,I8,58H   (REFINEMENT LEVEL TO WHICH CON
     ASTANT DENSITY IS ASSUMED))

      LINES = 64
      CALL FHEAD(13)
      WRITE (6,1090) FARFLD
      WRITE (6,1092) IADM,RHOBAS,RHOAMP,TOLRL
 1090 FORMAT (/26H SPECIAL BOUNDARY OPTIONS-/ 6X,7HFARFLD=,2(2X,A6))
 1092 FORMAT(/ 28H MATRIX SOLUTION PARAMETERS-/6X,7HIADM  =,I8,3X,70H(=-
     11,0,1, FOR STREAMLINE, ALTERNATING, AND ORTHOGONAL LINE RELAXATION
     2)/ 6X,7HRHOBAS=,F8.3,3X,33H(ACCELERATION FACTOR, BASE LEVEL)/
     36X,7HRHOAMP=,F8.3,3X,45H(ACCELERATION FACTOR, AMPLITUDE OF VARIATI
     40N)/  6X,7HTOLRL =,  E8.1,3X,30H(TOLERANCE RELATIVE TO MAXDS2)  )

C     PRINT HIGHLIGHT AND MAX. BODY RADII AND AREAS
      AHL    = RHL
      IF(AXIA) AHL=PI*RHL*RHL
```

```
      ARM    = RM
      IF(AXIA) ARM=PI*RM*RM
      WRITE (6,1091) RHL,AHL,RM,ARM
 1091 FORMAT (//6X, 17HHIGHLIGHT RADIUS=,F8.3,4X,15HHIGHLIGHT AREA=,
     * F8.3/6X,17HMAX. BODY RADIUS=, F8.3,4X,15HMAX. BODY AREA=,F8.3)
C     PRINT CHANNEL TABLE OF CONTENTS
      CALL PHEAD(2)
      WRITE (6,1060)
      LH     = LHO
   80 IF(LH.GE.LHE) GO TO 96
      MOREL = 4
      IF(NR(LH).NE.0) MOREL=MOREL+2+NR(LH)
      CALL PHEAD(MOREL)
      LH2    = LH+9
      WRITE (6,1070) CHNAM(LH),(WTFLOW(LHX),LHX=LH,LH2)
      NCX    = NC(LH)
      IF(NR(LH).LE.0) GO TO 95
      WRITE (6,1080) (TAB(I),I=1,NCX)
      CALL TABPRT(2HB=,BB(LH),NCX*NR(LH),NCX)
   95 LH     = LH+LHNEXT(LH)
      GO TO 80
   96 CONTINUE
 1060 FORMAT(/1X26HCONTENTS OF CHANNEL TABLE-)
 1070 FORMAT(//6X7HCHN    =2X,A6,5X7HWTFLOW=  E12.4,/6X7HTTO    =  F8.2,5X
     *7HPTO    =F8.3,5X7HTSO    =F8.2,5X7HPSO    =F8.3,/6X7HMACHO =F8.4,5X7
     *HAO     =  E12.4,1X7HVARY  =L8,/6X7HRG     =  F8.2,5X7HGAM    =F8.4,)
 1080 FORMAT(/6X7HNB/TAB=2X,A6,1H,5X,A6,1H,5X,A6,1H,5X,A6,1H,5X,A6,1H,)

C     LOOP THROUGH CHANNELS TO PRINT FLOW RATES, PRESSURES, AND TEMP
      RHOINF= PSA/(RGA*TSA)
      VINF   = SQRT(GAMA*RGA*TSA)*MACHA
      WTNORM= RHOINF*VINF*PI
      J2     = 0
      IC     = 0
  100 J2     = J2+1
      JCHN   = SLCHN(J2)
  105 IF(JCHN.NE.SLCHN(J2+1) .OR. J2.EQ.NJ) GO TO 110
      J2     = J2+1
      GO TO 105
  110 IC     = IC+1
      WTFA(IC)=W(J2)/WTNORM
      IF( RGA.NE.1. ) WTFA(IC)=W(J2)
      ICHN(IC)=JCHN
      LT     = LTO
  115 IF(JCHN.EQ.CH(LT)) GO TO 120
      LTP    = LT+LTNEXT(LT)
      IF(LTP.GE.LTE) GO TO 120
      LT     = LTP
      GO TO 115
  120 LTP    = LT+LPSI(LT)+NPT(LT)-1
      WTFS(IC)=XCH(LTP)/WTNORM
      IF( RGA.NE.1. ) WTFS(IC)=XCH(LTP)
      WPTO(IC)=PTC
      WTTO(IC)=TTC
      LH     = LHO
  122 IF(JCHN.EQ.CHNAM(LH)) GO TO 124
      LHP    = LH+LHNEXT(LH)
      IF(LHP.GE.LHE) GO TO 128
```

```
*DECK WRIA
      SUBROUTINE WRIA
*WRIA--          WRITE THE KEY(5)=A  STC DATA RECORD          -WRIA-

      COMMON /ALLCOM/ MACHA,PSA,TSA,PTA,TTA, AXIA,RGA,GAMA,
     1                MACHC,PSC,TSC,PTC,TTC, AXIC,RGC,GAMC,
     2                DAXIT,SCALEA,TTE,CHOTST
       REAL           MACHA(1),MACHC
       LOGICAL        AXIA,AXIC
       LOGICAL        CHOTST
       REAL           MACHO
       EQUIVALENCE    (MACHO,MACHA),(PSO,PSA),(TSO,TSA),
     1                          (AXI,AXIA),(RG,RGA),(GAM,GAMA)
      COMMON /CSS    / SSFML,SSEF,SSEANG,SSDF,SSFEND,SSFND1
     1               ,SSDLE,A4FACT,BRLX,CURRLX,TSIC,RHOC,RHOCSS
       INTEGER        SSFML
       LOGICAL              SSEF,       SSDF,   SSDLE
C      SSFML = SUPERSONIC CURVATURE FORMULA NUMBER
C      SSEF  = SUPERSONIC ENTERING FLOW, T OR F
C      SSEANG= ENTERING FLOW ANGLE (DEGREES) FOR SSEF=T
C      SSDF  = SUPERSONIC DISCHARGE FLOW, T OR F
C      SSFEND= SUPERSONIC BEAM DOWNSTREAM END CONDITION, =0,1 FOR PARABOL
C      SSFND1= SUPERSONIC BEAM UPSTREAM END CONDITION, =0,1, FOR PARABOLA
C      SSDLE = SS FLOW BELOW AND AFT OF LE PT, T OR F
C      A4FACT= CENTRAL POINT INFLUENCE COEFFICIENT FACTOR
C      BRLX  = B-RELAXATION FACTOR
C      CURRLX= CURVATURE RELAXATION FACTOR
C      TSIC  = NUMBER OF POINTS TO BE READ FOR TRANSONIC INTERPOLATION
      COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
     *                LO,LESTA,  LDUM(8),
     *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
     *                LEO,LEE, LRO,LRE,LRD
       DIMENSION      LIMITS(24)
       EQUIVALENCE    (LIMITS,LHO)
      COMMON /SLTAB / W(128),X2(128),SLCHN(128)
       INTEGER SLCHN

      COMMON /BCOMMN/ PROGM(9),FILIN,FILOT
       LOGICAL        FILIN, FILOT
      COMMON /ADAM01/ NAME(6),ADDRES(6),TITLE(6),IDENT(6)
      COMMON /BENDIN/ NBCIN(2),ACF(2)
      COMMON /CB     / B(300)
      COMMON /CBITS / BITS,IBLANK
      COMMON /CCRX   / CRXSL,CRXOL,CRXSS,CRXE,CRXC,CRMACH
       DIMENSION      CRX(6)
       EQUIVALENCE    (CRX,CRXSL)
C      CHANNEL INPUT DATA TABLE
C      INDEX - LH=LHO,LHE
      COMMON /CHDATA/ CHNAM(1),LHNEXT(1),WTFLOW(10),NR(1),NC(1),TAB(6),
     1                BB(75)
       DIMENSION      TABLES(998), TTO(1),PTO(1)
C     TABLE OF CONVECTED PROPERTIES
C      INDEX- LT=LTO,LTE
C      CH     = CHANNELNAME
C      LTNEXT= INDEX INCREMENT TO THE NEXT CHANNEL
C      LPSI   = RELATIVE LOCATION OF PSI LIST
C      NPT    = NO. OF PSI, TT, PT AND RCU VALUES
       DIMENSION      CH(1),LTNEXT(1),NPT(1),LPSI(1),LTT(495)
```

```
      INTEGER CH,CHNAM
      DIMENSION XCH(1)
      EQUIVALENCE     (CHNAM,TABLES,CH,XCH),(LHNEXT,LTNEXT),
     *                (WTFLOW,NPT),
     *                (WTFLOW(2),LPSI,TTO),
     *                (WTFLOW(3),LTT,PTO)
      COMMON /CIADIN/ RHOBAS,RHOAMP,IADM
      COMMON /CINNER/ INRCTR,RDUM,NINNER(16),CNVF(16)
      COMMON /CISBOT/ FARFLD(2),FREE(2),PRES(2),RFF,NZP,
     1                ZP(10),PPS(10), A1,A2,ADUM(6)
      INTEGER         FARFLD,FREE,PRES
      COMMON /CLINES/ LINES,OMTIFK,PTITLE(6)
      LOGICAL               OMTIFK
      COMMON /CM     / JMS(300)
      COMMON /CMAXIT/ MAXIT,MAJCTR,GREFIN,EDUM
      LOGICAL                     GREFIN
      EQUIVALENCE (MAJCTR,NREFIN)
      COMMON / CNORM / RHL,RM,AHL,ARM
      COMMON /CPI    / PI,TWOPI,PIQ2,PIQ4,TODEG,TORAD
      COMMON /CPRINT/ PDUM1(3),PREFIN,PREFN2,PDUM(11)
      COMMON /CPRPRN/ PRPRN
      INTEGER         PRPRN
      COMMON /CPTMOV/ VELPOT,ICOB,NODENS,CPTDUM
      LOGICAL         VELPOT
      COMMON /CR     / RF(300)
      COMMON /CREFIN/ SLS,SG21,VMG1,VMG2, NGR,NGZ,SGR(10),GR(10),
     1                SGZ(10),GZ(10)
      COMMON /CS1    / S1(300)
      COMMON /CS2    / S2(300)
      COMMON /CTOLRL/ TOLRL,MAXSWP,CLEN,DS2MX,TOLES2,NSWP,
     1                DS1DMP,DS1MXA,DS1MXB,DS1RMS,ES2MX,DS1RMO,
     2                SG1MIN,TOLINR
      COMMON /CTHICK/ NTHKX,NTHKY,THKX(20),THKY(20),THIK2D(78)
      COMMON /CVM    / VMF(300)
      COMMON /CZ     / ZF(300)

      COMMON /CHNFPT/ ICHN(10),WTFS(10),WTFA(10),WPTO(10),WTTO(10), IC
      COMMON /TAPES / NTAPO,NTAPN

      LOGICAL STCFIL
      DATA STCFIL/T/
      DATA KA/1HA/


      OMTIFK= .TRUE.
      IF(FILOT) OMTIFK=.FALSE.
      CALL FHEAD(64)
      TSC    = TSA
      TTC    = TSC*(1.+(GAMA-1.)*.5*MACHO*MACHO)
      PTC    = PSC*(TTC/TSC)**(GAMA/(GAMA-1.))
   55 WRITE(6,1000) AXI,MACHO,RG,TSC,GAM,PSC,TTE,PTC,CHOTST,TTC,
     1                NBCIN,ACF
 1000 FORMAT (/15H GENERAL INPUT-// 6X,7HAXI    =,L8,26X,7HMACHO =,F8.4/
     16X,7HRG     =,F8.2,26X,7HTSO    =,F8.2/ 6X,7HGAM    =,F8.4,26X,
     17HPSO    =,F8.3/ 6X,7HTTE    =,F8.3,26X,7HPTO    =,F8.2/ 6X,7HCHOTST=
     1,L8,26X,7HTTO    =,F8.3//  27H STREAMLINE END CONDITIONS-/ 6X,7HNBC
     1IN =,218/ 6X,7HACF    =,2F8.3/ )
```

```
*DECK STALOO
      SUBROUTINE STALOO
*STALOO         LOOP THROUGH STATIONS AND EXECUTE FLOBAL     -STALOO-

       COMMON /CFB    / L,MA,MB,PLB,PUB,WF,CHOKE,SUBSON, NK,PLBC,PUBC,
      1                 XCHOKE, TAREA,VMBC, WRQST,WCALC, QV(8),QVP(8),
      *                 JSUM,VMLBSQ
        LOGICAL                              CHOKE,SUBSON
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                 LO,LESTA,  LDUM(8),
      *                 MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                 LEO,LEE, LRO,LRE,LRD
        DIMENSION       LIMITS(24)
        EQUIVALENCE     (LIMITS,LHO)
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
       COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
      1                 TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
      1                 TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
      3                 VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
        LOGICAL         PRIM
        INTEGER TYPELB,TYPEUB
        DIMENSION       SCHOKE(1)
        EQUIVALENCE     (SCHOKE,DWDV)

       COMMON /CB     / B(300)
       COMMON /CFB2   / PASS1
        LOGICAL         PASS1
       COMMON /CINNER/ INRCTR,RDUM(33)
       COMMON /CPRINT/ PRTES2,PRTB,PRTA,PREFIN,PREFN2,SSONIC,PDUM(10)
       COMMON /CRHS   / RHS(300)
       COMMON /CTOLRL/ TOLRL(10),ES2MX,DS1RMO,SG1REF,TOLINR

C     BEGIN LOOP THROUGH STATIONS
       CHOKE = .FALSE.
       JSUM  = 0
       L     = LO

C     CALL BRHS AND FLOBAL
  410  PLB   = 0.
       PUB   = 0.
       WF    = 0.
       CALL BRHS

C     INDEX TO THE NEXT STATION (I.E. ORTHOGONAL)
  450  L     = L+LNEXT(L)
       IF(L.LT.LESTA) GO TO 410

       PASS1 = .FALSE.
       RETURN
       END
       OVERLAY(STC,2,2)
```

```
*DECK STCWI
      PROGRAM STCWI
C     WRITE THE OVER-ALL STC DATA RECORD, KEY(5)=A.
      CALL WR1A
      RETURN
      END
```

```
*DECK NEWRAP
      SUBROUTINE NEWRAP(X,E,V)
*NEWRAP     OUTSIDE ITERATION PROCEDURE                          -NEWRAP-
C           TO BE USED WHEN INNER SELF CONVERGENT RELATIONS EXIST.

C     INPUT-
C      X      - ABSCISSA
C      E      - ERROR IN THE ORDINATE
C      V      - STORAGE FOR A 12 ELEMENT VECTOR
C     INPUT, FIRST ENTRY ONLY
C      V(1)   = CTR    = 0.
C      V(2)   = DEDX   = ESTIMATE OF THE SLOPE OF THE CURVE
C               (X2=X1-E1/DEDX  IS THE FORMULA FOR THE SECOND X)
C               (E/DEDX)  IS USED TO REDUCE DXMAX DURING THE ITERATION
C      V(3)   = XMOVE
C         ABS(XMOVE) = MAXIMUM DELTA X
C         SIGN(XMOVE)= DIRECTION TO THE BRANCH OF THE CURVE WITH SLOPE=SI

C     OUTPUT-
C      X      = NEXT X ESTIMATE

       COMMON /CNEWR / DEDXP(2),DXP(2),DX,WS

       DIMENSION   V(12),Q(12), XP(2),EP(2)
       EQUIVALENCE (CTR,Q(1)), (DEDX,Q(2)), (XMOVE,Q(3)),
      1            (DXMAX,Q(5)), (DXPREV,Q(6)), (OPSIGN,Q(7)), (SPAN,Q(8)
      2),           (XP,Q(9)), (EP,Q(11))

       LOGICAL SPAN

       DO 50 I = 1,12
    50 Q(I)  = V(I)
       IF(CTR.GE.30.) CALL ERROR1
       IF(CTR.NE.0.) GO TO 200

C     FIRST ENTRY
       DX     = -E/DEDX
       DXMAX  = ABS(XMOVE)
       DXPREV= DXMAX
       OPSIGN= 0.
       SPAN   = .FALSE.
       GO TO 520

C     SECOND AND SUCCESSIVE ENTRIES, EVALUATE DEDXP(I) AND DXP(I)
   200 WS     = 0.
       DO 250 I=1,2
       DXP(I)= 0.
       IF(I.EQ.1) GO TO 220
       IF(CTR.LE.1.) GO TO 270
       IF(WS.EQ.0.) GO TO 220
       IF(.NOT.SPAN .OR. (E*EP(2).GT.0.)) GO TO 250
C         IF(.NOT.SPAN .OR. SAMESIGN(E,EP(2)))  DO NOT USE POINT 2
   220 DE     = E-EP(I)
       DX     = X-XP(I)
       IF(ABS(DE).LT.ABS(DX)/1.E15) GO TO 250
       IF(ABS(DX).LT.ABS(DE)/1.E15) GO TO 250
       DEDXP(I)= DE/DX
C         CHECK SIGN OF DEDXP(I)
```

```
      IF(DEDXP(I)*DEDX.LT.0.) GO TO 250
      DXP(I)= AMAX1(-DXMAX,AMIN1(-E/DEDXP(I),DXMAX))
      WS    = WS+1.
  250 CONTINUE
  270 IF(WS.NE.0.) GO TO 400

C     THE DEDXP HAVE INCORRECT SIGNS
C     TAKE MAX JUMP TOWARD THE CORRECT BRANCH
C     MAYBE DESIRED ORDINATE IS ABOVE/BELOW THE MAX/MIN OF THE CURVE
  350 IF(OPSIGN) 360,360,355
  355 DXMAX = .5*DXMAX
  360 OPSIGN= -1.
      DX    = XMOVE
      GO TO 520

C     REDUCE MAX DX IF DIRECTION OF ITERATION IS CHANGING
  400 IF(OPSIGN) 410,490,490
  410 DXMAX = .5*DXMAX
  490 OPSIGN= 1.

C     PEDICT NEXT ABSCISSA,  DEDXP HAVE THE CORRECT SIGNS
  500 DX    =(DXP(1)+DXP(2))/WS
      DXMAX = AMIN1(DXMAX,ABS(XMOVE))
C      -DXMAX.LE.DX.LE.DXMAX
  520 DX    = AMAX1(-DXMAX,AMIN1(DX,DXMAX))

C     SAVE CERTAIN GOODIES TO USE FOR FUTURE ENTRIES
  600 DXMAX = .25*DXMAX + .75*AMIN1(DXMAX,AMAX1(DXPREV,ABS(2.*E/DEDX)))
      DXPREV= ABS(DX)
      XP(2) = XP(1)
      EP(2) = EP(1)
      XP(1) = X
      EP(1) = E
      IF(EP(1)*EP(2).LT.0.) SPAN=.TRUE.
      CTR   = CTR+1.

C     SET X AND RETURN
      X     = X+DX
      DO 960 I=1,12
  960 V(I)  = Q(I)
      RETURN
      END
```

```
          M      = M+1
          CALL GETIX
          ISTAG = 3
          CALL SAVIX
          RHS(M)= 0.
          B(M)   = BDUMMY
          GO TO 756

C     SPECIAL BOUNDARY TYPES
  724 IF(TYPELB(L).NE.FARFLD .AND. TYPELB(L).NE.FREE .AND.
     *    TYPELB(L).NE.PRES) GO TO 756
       B(M)   = .5*(AREAO(2)-AREAO(1))*BETSQP*(S2(2)-S2(1))
       RHS(M)= AREA(2)-AREAO(2) - AREA(1)+AREAO(1)
       IF(VMLBSQ.NE.0.) RHS(M)=RHS(M)
      1         -(AREAO(2)-AREAO(1))*BETSQP*.5*(VMLBSQ/(VM(M)*VM(M))-1.)
       GO TO 756

C     INTERIOR POINT
  725 BM     = 0.
       IF(MM.NE.M) GO TO 726
       TSAVGM= .5*(TS(KM)+TS(KM1))
       QGAMM = FGR(KM)/(1.+FGR(KM))
       BETSQM= 1.-VM(MM)*VM(MM1)*QGAMM/(RG(KM)*TSAVGM)
       RHOVM = .5*(WQA(KM1)+WQA(KM))
       BM     = .5*BETSQM*(S2(MM)-S2(MM1))/RHOVM
  726 IF(WSTA(K+1).EQ.WSTA(K)) GO TO 728
       TSAVGP= .5*(TS(K)+TS(K+1))
       QGAMP = FGR(K)/(1.+FGR(K))
       BETSQP= 1.-VM(M)*VM(M+1)*QGAMP/(RG(K)*TSAVGP)
       RHOVP = .5*(WQA(K+1)+WQA(K))
       BM     = .5*BETSQP*(S2(M+1)-S2(M))/RHOVP + BM
  728 IF(SLSWI(L).NE.0.) GO TO 730
       B(M)   = BM
       GO TO 732
  730 B(M)   = BRLX*BM+(1.-BRLX)*B(M)
  732 IF(MM.EQ.M .AND. B(M)*B(M-1).LT.0.) SSOL=.TRUE.
       IF(WSTA(K+1).EQ.WSTA(K)) GO TO 757
  735 RHS(M)=(AREA(K+1)-AREAO(K+1)-AREA(K)+AREAO(K))/(WSTA(K+1)-WSTA(K))
      1       -(AREA(KM)-AREAO(KM)-AREA(KM1)+AREAO(KM1))/(WSTA(KM)-WSTA(KM
      2         1))
  756 KM1    = K
       MM1    = M
       KM     = KM1+1
       MM     = MM1+1
       GO TO 760

C     DOUBLE POINT (I.E. W(K+1)=w(K))
  757 RHS(M)= ES2(K)-ES2(K+1)

  760 K      = K+1
       M      = M+1
       IF(K.LT.NK) GO TO 725

C     UPPER BOUNDARY
C      NOTE- MB=MUB(L)-1  FOR A STAGNATION POINT
       M      = MUB(L)
       RHS(M)= 0.
       M      = MB
```

151

```
      RHS(M)= 0.
      QGAMM = FGR(K)/(1.+FGR(K))
      BETSQM= 1.-VM(M)*VM(M)*QGAMM/(RG(K)*TS(K))
      B(M)  = BETSQM*(S2(M)-S2(M-1))/WQA(K)
      IF(B(M).EQ.0.) B(M)=BDUMMY

C     SPECIAL BOUNDARY TYPES
C     TEST FOR SHOCK ON ORTHOGONAL
      CALL GETIX
      IF( MU.EQ.0 .AND. MD.EQ.0 ) GO TO 800
      IF(TYPEUB(L).NE.PRES .AND.
     1   TYPEUB(L).NE.FREE .AND.
     2   TYPEUB(L).NE.FARFLD) GO TO 800
      B(M)  = .5*(AREAO(K)-AREAO(K-1))*BETSQM*(S2(M)-S2(M-1))
      RHS(M)= AREA(K-1)-AREAO(K-1) - AREA(K)+AREAO(K)
  800 IF((B(M)*B(M-1)).LT.0.) SSOL=.TRUE.
C.....END CALC OF  B  AND  RHS

      SLSWI(L)=0.
      IF(SSOL) SLSWI(L)=1.
      RETURN
      END
```

```
      IF(SCHOKE(L).NE.XCHOKE) GO TO 500
      IF(SSDF) SUBSON=.FALSE.
      JSUM  = JA+256*JB

C     EXECUTE FLOW BALANCE
  500 CALL FLOBAL
      IF(TYPELB(L).EQ.TE .OR. TYPEUB(L).EQ.TE) JSUM=0
      IF( MA.EQ.MB ) RETURN
      VMB(L)= VMBC

C     EVALUATE S2-DEVIATIONS
      F     = 1.
      IF((TYPELB(L).EQ.SOLID .AND. TYPEUB(L).EQ.SOLID) .OR.
     *  TYPELB(L).EQ.FIELD .OR. TYPEUB(L).EQ.FIELD) F=AREAO(NK)/AREA(NK)
      IF(SSONIC.EQ.2.) F=1.
C     (PLANE 2-D)
      DO 510 K=1,NK
  510 ES2(K)= (F*AREA(K)-AREAO(K))/LAMBDA(K)
      IF(.NOT.AXIA) GO TO 550
C     (AXISYMMETRIC)
      K     = 2
      M     = MA+1
  520 ES2(K)= ES2(K)/(TWOPI*R(M))
      K     = K+1
      M     = M+1
      IF(K-NK) 520,520,550


C     EVALUATE MAXIMUM FLOW BALANCE ERROR, ES2MX
  550 IF(L.EQ.LO) ES2MX=0.
      DO 560 K=1,NK
  560 ES2MX = AMAX1(ES2MX,ABS(ES2(K)))

      IF (PRTES2 .LE. 2.) GO TO 600
      IF (X1(L) .LT.PDUM(8).OR.X1(L).GT.PRTES2) GO TO 722
      LMX1    = L
      LMX2    = L
      NKX1    = NK
      CALL MOVE (1,ES2,ES2X1,NK,1)
      IF (X1(L).EQ.PDUM(8)) WRITE(6,1661)
      GO TO 660

  600 IF (PRTES2.NE.2.) GO TO 722
      DATA ENTRY2/F/
      ES2MX0=0.
      DO 605 K=1,NK
  605 ES2MX0= AMAX1(ES2MX0,ABS(ES2(K)))
      IF (ENTRY2) GO TO 610
      ES2MX1= ES2MX0
      ES2MX2= ES2MX0
      LMX1 = L
      LMX2 = L
      NKX1    = NK
      NKX2    = NK
      CALL MOVE (2,ES2,ES2X1,NK,1, ES2,ES2X2,NK,1)
      ENTRY2 = .TRUE.
      GO TO 690
  610 IF(ES2MX0.LE.ES2MX1) GO TO 630
```

```
      ES2MX2 = ES2MX1
      LMX2 = LMX1
      NKX2   = NKX1
      CALL MOVE (1,ES2X1,ES2X2,NKX1,1)
      ES2MX1= ES2MX0
      LMX1 = L
      NKX1   = NK
      CALL MOVE(1,ES2,ES2X1,NK,1)
      GO TO 650
  630 IF(ES2MX0.LE.ES2MX2) GO TO 650
      ES2MX2= ES2MX0
      LMX2 = L
      NKX2   = NK
      CALL MOVE(1,ES2,ES2X2,NK,1)
  650 IF(MBB.NE.NM) GO TO 690
      WRITE (6,1661)
  660 WRITE(6,1660) X1(LMX1)
      M = MLB(LMX1)-1
      IF(LMX1.EQ.L) M=MA-1
      DO 670 K=1,NKX1
      M = M+1
  670 WRITE(6,1670)B(M),RHS(M),DS2(M),Z(M),R(M),PHI1(M),CURV(M),ES2X1(K)
      IF(LMX1.EQ.LMX2) GO TO 690
      LMX1 = LMX2
      NKX1   = NKX2
      CALL MOVE (1,ES2X2,ES2X1,NKX2,1)
      GO TO 660
 1661 FORMAT(1H1)
 1660 FORMAT (//9H STATION=,F8.3//5X,1HB,10X,3HRHS,9X,3HDS2,9X,1HZ,10X,
     1        1HR,10X,4HPHI1,7X,4HCURV,7X,5HES2X1/)
 1670 FORMAT (1X,F11.5,2(3X,F9.6),4(F11.5  ),3X,F9.6)
  690 CONTINUE

C****CALC COEFICIENT  B  AND  RHS  OF MATRIX EQUATION FOR DS2

C     SET SUPERSONIC OL INDICATOR
  722 SSOL   = .FALSE.

C     LOWER BOUNDARY
C      NOTE- MA=MLB(L)+1  FOR A STAGNATION P/INT
      M      = MLB(L)
      RHS(M)= 0.
      K      = 1
      M      = MA
      RHS(M)= 0.
      QGAMP = FGR(1)/(1.+FGR(1))
      BETSQP = 1.-VM(M)*VM(M)*QGAMP/(RG(1)*TS(1))
      B(M)   = BETSQP*(S2(M+1)-S2(M))/WQA(1)
C       IS FIRST POINT AN ISTAG=3 PT AND THE FIRST OF A DOUBLE POINT
        IF(WSTA(2).NE.WSTA(1)) GO TO 724
        IF(TYPELB(L).NE.FIELD) CALL ERROR1
C       TREAT FIRST PT AS DUMMY PT AND 2ND PT AS ISTAG=3 PT
        RHS(M)= ES2(1)-ES2(2)
        B(M)   = BDUMMY
        CALL GETIX
        ISTAG = 0
        CALL SAVIX
        K      = K+1
```

154

```
*DECK BRHS
      SUBROUTINE BRHS
*BRHS--          COEFFICIENT B AND RHS TERMS                    -BRHS-

C     OUTPUT-
C      RHS(M)= RIGHT HAND SIDE OF THE MATRIX EQUATION FOR  DS2
C      B(M)  = COEFFICIENT OF THE CURVATURE TERM

       COMMON /ALLCOM/ MACHA,PSA,TSA,PTA,TTA, AXIA,RGA,GAMA,
      1                MACHC,PSC,TSC,PTC,TTC, AXIC,RGC,GAMC,
      2                DAXIT,SCALEA,TTE,CHOTST
        REAL           MACHA(1),MACHC
        LOGICAL        AXIA,AXIC,CHOTST
       COMMON /CFB    / L,MA,MB,PLB,PUB,WF,CHOKE,SUBSON,NK,PLBC,PUBC ,
      1                XCHOKE, TAREA,VMBC, WROST,WCALC, QV(8),QVP(8),
      *                JSUM,VMLBSQ
        LOGICAL                              CHOKE,SUBSON
       COMMON /CSS    / SSFML,SSEF,SSEANG,SSDF,SSFEND,SSFND1
      1                ,SSDLE,A4FACT,BRLX,CURRLX
        INTEGER        SSFML
        LOGICAL              SSEF,       SSDF,   SSDLE
C      SSFML = SUPERSONIC CURVATURE FORMULA NUMBER
C      SSEF  = SUPERSONIC ENTERING FLOW, T OR F
C      SSEANG= ENTERING FLOW ANGLE (DEGREES) FOR SSEF=T
C      SSDF  = SUPERSONIC DISCHARGE FLOW, T OR F
C      SSFEND= SUPERSONIC BEAM DOWNSTREAM END CONDITION, =0,1 FOR PARABOL
C      SSFND1= SUPERSONIC BEAM UPSTREAM END CONDITION, =0,1, FOR PARABOLA
C      SSDLE = SS FLOW BELOW AND AFT OF LE PT, T OR F
C      A4FACT= CENTRAL POINT INFLUENCE COEFFICIENT FACTOR
C      BRLX  = B-RELAXATION FACTOR
C      CURRLX= CURVATURE RELAXATION FACTOR
       COMMON /ERASE2/ AREA(96),AREA0(96),DISP(96),PT(96),LAMBDA(96),
      1                RHO(96),SQRTVV(96),TS(96),TT(96),VMSQ(96),
      2                VVKQKP(96),
      2                WQA(96),WSTA(96), RG(96),C2CP(96),FGR(96)
        REAL           LAMBDA
        DIMENSION      ES2(96),SDNQRM(96)
        EQUIVALENCE    (ES2,VVKQKP),(SDNQRM,RHO)
        DIMENSION      RCU(96)
        EQUIVALENCE    (RCU,LAMBDA)
C      INDEX- M=MO,NM
       COMMON /CZ     / Z(300)
       COMMON /CR     / R(300)
       COMMON /CS2    / S2(300)
       COMMON /CS1    / S1(300)
       COMMON /CPHI1  / PHI1(300)
       COMMON /CM     / JMS(300)
       COMMON /CCURV  / CURV(300)
       COMMON /CB     / B(300)

       COMMON /CIDEX  / M,J,MU,MD,ISTAG
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, IRO,LRL,LRD
        DIMENSION      LIMITS(24)
        EQUIVALENCE    (LIMITS,LHO)
       COMMON /CBEND  / NBCB(2),ANGE(2),CURVE(2),FB(2)
```

```
      COMMON /CBITS / BITS,BLANK
      COMMON /CCUBE / NBC(2),C1(2),C2(2),FEND(2)
      COMMON /CGRAV / GG
      COMMON /CPI   / PI,TWOPI,PIO2,PIO4,TODEG,TORAD
      COMMON /CREFIN/ SG1,SG2,VMG1,VMG2
     1,               NGR,NGZ, SGR(10),GR(10), SGZ(10),GZ(10)
      COMMON /SLTAB / W(128),X2(128),SLCHN(128)
        INTEGER SLCHN
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
      COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
     1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
     1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
     3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
        LOGICAL         PRIM
        INTEGER TYPELB,TYPEUB
        DIMENSION       SCHOKE(1)
        EQUIVALENCE     (SCHOKE,DWDV)
        DIMENSION       SLSWI(1)
        EQUIVALENCE     (SLSWI,VCL)
C      SLSWI = SONIC LINE/SHOCK WAVE INDICATOR

      COMMON /CDS2  / DS2(300)
      COMMON /CMAXIT/ MAXIT,MAJCTR,GREFIN,EDUM
       LOGICAL                     GREFIN
      COMMON /CPRINT/ PRTES2,PRTB,PRTA,PREFIN,PREFN2,SSONIC,PDUM(10)
      COMMON /CRHS  / RHS(300)
      COMMON /CTOLRL/ TOLRL,MAXSWP,CLEN,DS2MX,TOLDS2,NSWP,
     1                DS1DMP,DS1MXA,DS1MXB,DS1RMS,ES2MX
      COMMON /CTABPR/ I1TAB
      COMMON /CVM   / VM(300)

        INTEGER          FARFLD,FIELD,FREE,PRES,SOLID,TE
      DIMENSION ES2X1(96),ES2X2(96)
       LOGICAL         ENTRY2,  SSOL
C      SSOL   = SUPERSONIC POINT ON THIS OL, T OR F

      DATA FARFLD/6HFARFLD/, FIELD/5HFIELD/, FREE/4HFREE/,
     *     PRES/4HPRES/, SOLID/5HSOLID/, TE/2HTE/

C     INITIALIZE
      TOLS2 = AMAX1(.01,2.*TTE)
      BDUMMY= 1./1024.

C     SUBSONIC/SUPERSONIC BRANCH SELECTION
      M      = MLB(L)
      CALL GETIX
      JA     = J
      MAA    = M
      M      = MUB(L)
      CALL GETIX
      JB     = J
      MBB    = M
      IF(JSUM.EQ.0) SUBSON=.TRUE.
      IF(SSEF) SUBSON=.FALSE.
```

```
      GO TO 255

C     FLOW IS NOT CHOKED
  245 L      = LX2
      PLB    = PLBX
      PUB    = PUBX
      CALL FLOBAL
  250 VMB(L)= VMBC
      NLF    = (LFE+1-LFO)/NFCOLS
      ILF    = (LF-LFO)/NFCOLS
      IF(INRCTR.EQ.0 .OR. MOD(INRCTR-1,NLF).NE.ILF) GO TO 290
      IF(VNR(LF).NE.0.) GO TO 252
      VNR(LF+1)=-2.
      VNR(LF+2)=.25*WRQST
  252 WNEW   = WRQST
      VNR(LF+6)=0.
      CALL NEWRAP(WNEW,WCALC-WRQST,VNR(LF))
      IF(VNR(LF+6).EQ.(-1.)) WNEW=WCALC
      RATIO = WNEW/WRQST

C    ADJUST FLOW IN THE STREAMLINE TABLE
  255 M      = MLB(L)
      CALL GETIX
      JA     = J
      M      = MUB(L)
      CALL GETIX
      JB     = J
C      CHECK TO SEE IF USER WISHES FLOW RATE TO BE VARIED
      JX     = JA
  258 LH     = LHO
  260 IF(LH.GE.LHE) GO TO 267
      IF(CHNAM(LH).EQ.SLCHN(JX)) GO TO 265
      LH     = LH+LHNEXT(LH)
      GO TO 260
  265 IF(.NOT.VARY(LH)) GO TO 280
  267 IF(JX.EQ.JB) GO TO 270
      JX     = JB
      GO TO 258
C      ADJUST FLOWS
  270 DO 275 J=JA,JB
  275 W(J)   = W(J)*RATIO
      GO TO 290
C      DO NOT ADJUST FLOWS, PRINT COMMENT IF SUPER-CHOKED
  280 IF(SCHOKE(LK2).NE.XCHOKE) GO TO 290
      IF(RATIO.LT.1.) GO TO 282
      SCHOKE(LK2)=0.
      GO TO 290
  282 WRITE (6,1280) RATIO,X1(LK2),CHNAM(LH)

C    S1-COORDINATE ON UPPER SURFACE AT THE T.E.
  290 IF(JORDER(LF).LT.0) GO TO 295
      M      = MLB(LXA)
      S1F(LF)=S1(M)

C    INDEX TO THE NEXT T.E.FLOW ADJUSTMENT X1-STATIONS
  295 LF     = LF+NFCOLS
      GO TO 101
```

157

```
C     ALL FLOW ADJUSTMENTS COMPLETED,
C     RETURN FOR FLOW BALANCE AT ALL STATIONS.
  300 IF(PDUM(6).EQ.0.) RETURN
      IITAB = LFO
      CALL TABPRT(6HCADJWF,X1F,LFE,10)
      CALL TABPRT(1HW,W,NJ,10)
      RETURN

 1280 FORMAT(/65H ***  THE CHOKED FLOW RATE IS LESS THAN THE USER INPUT
     *FLOW/RATE. ,6X,8HRATIO = ,F9.6,3X,6HSTA = ,F8.3,3X,6HCHN = ,A6)
      END
```

```
      IF(X1BF(LF).NE.X1TE) CALL STAX1(X1BF(LF),X2TE,-1.,LKB,DUM)
      IF(X1AF(LF).EQ.X1TE) GO TO 120
  118                      CALL STAX1(X1AF(LF),-1.,X2TE,DUM,LKA)
  120 IF(JORDER(LF)) 130,140,200

C     SINGLE CHANNEL CHOKE
  130 CHOKE = .TRUE.
      L     = LKA
      CALL FLOBAL
      SCHOKE(L)=XCHOKE
      LK2   = L
      VMB(L)= VMBC
      RATIO = WCALC/WRQST
      GO TO 255

C**   ITERATE FOR T.E. PRESSURE, JORDER(LF)=0
  140 IF(.NOT.CHOTST) GO TO 150
      L     = LKB
      CHOKE = .TRUE.
      CALL FLOBAL
      PUBX  = PUBC
      WBCHOK= WCALC
      L     = LKA
      CALL FLOBAL
      PLBX  = PLBC
      WACHOK= WCALC
      CHOKE = .FALSE.


  150 QVP   = 0.
      PSTE  = -1.
  155 L     = LXB
      PLB   = 0.
      PUB   = PSTE
      CALL FLOBAL
      VMBSAV= VMBC
      WBO   = WRQST
      WB    = WCALC

      L     = LXA
      IF(QVP.EQ.0.) PSTE=PUBC
      PLB   = PSTE
      PUB   = 0.
      CALL FLOBAL
      YO    = WBO+WRQST
      WAB   = WB+WCALC
      IF(CHOTST) WAB=AMIN1(WBCHOK,WB)+AMIN1(WACHOK,WCALC)
      YTOL  = 1.E-5*YO
      DYDX  = -1.E-5

      CALL QIREM(PSTE,WAB,.5*(PT(1)-PSTE),QVP)
      IF(QVP.NE.0.) GO TO 155
      VMB(LXB)=VMBSAV
      VMB(LXA)=VMBC

C     SETUP TO ADJUST FLOWS
      IF(NCHB(LF).NE.1) GO TO 170
C      NCHB(LF).EQ.1
```

159

```
      L      = LXB
      RATIO = WB/WBO
      GO TO 255
C      NCHA(LF).EQ.1
  170 RATIO = WCALC/WRQST
      GO TO 255

C**  CALCULATION OF TE PRESSURE (GIVEN FLOW) AT STATION LX1
  200 IF(JORDER(LF).EQ.2) GO TO 205
C      JORDER=1
      LX1    = LXB
      LX2    = LXA
      LK1    = LKB
      LK2    = LKA
      GO TO 210
C      JORDER=2
  205 LX1    = LXA
      LX2    = LXB
      LK1    = LKA
      LK2    = LKB
  210 L      = LX1
      CALL FLOBAL
      VMB(L)= VMBC
      IF(JORDER(LF).EQ.2) GO TO 220
      PLBX   = PUBC
      PUBX   = 0.
      GO TO 230
  220 PLBX   = 0.
      PUBX   = PLBC

C    CALCULATION OF FLOW (GIVEN TE PRESSURE) AT STATION LX2
  230 IF(.NOT.CHOTST) GO TO 245
C      CALCULATE MAXIMUM/CHOKED FLOW
      L      = LK2
      CHOKE = .TRUE.
      CALL FLOBAL
      CHOKE = .FALSE.
      VMBSAV= VMBC
      RATIO = WCALC/WRQST

C    CALCULATE PRESSURE AT THE T.E. STATION
  235 IF(LK2.EQ.LX2) GO TO 240
      L      = LX2
      WF     = WCALC
      IF(SSDF) SUBSON = .FALSE.
      CALL FLOBAL
      WF     = 0.
      SUBSON= .TRUE.
      NAMELIST /NLADJ/ LF,LX1,LK1,LX2,LK2,PLBX,PLBC,PUBX,PUBC
  240 IF(PDUM(6).EQ.2.) WRITE(6,NLADJ)
      IF((PLBX.NE.0. .AND. PLBC.GE.PLBX) .OR.
     1   (PUBX.NE.0. .AND. PUBC.GE.PUBX)) GO TO 242
      GO TO 245

C      CHOKED FLOW
  242 SCHOKE(LK2)=XCHOKL
      VMB(LK2)=VMBSAV
      VMB(L)= VMBC
```

```
*DECK ADJWF
      SUBROUTINE ADJWF
*ADJWF-         ADJUST WEIGHT FLOW                                  -ADJWF-

      COMMON /ALLCOM/ MACHA,PSA,TSA,PTA,TTA, AXIA,RGA,GAMA,
     1                MACHC,PSC,TSC,PTC,TTC, AXIC,RGC,GAMC,
     2                DAXIT,SCALEA,TTE,CHOTST
       REAL           MACHA(1),MACHC
       LOGICAL        AXIA,AXIC,CHOTST
      COMMON /CFB    / L,MA,MB,PLB,PUB,WF,CHOKE,SUBSON, NK,PLBC,PUBC,
     1                 XCHOKE, TAREA,VMBC, WRQST,WCALC, QV(8),QVP(8),
     *                 JSUM,VMLBSQ
       LOGICAL                          CHOKE,SUBSON
      COMMON /CSS    / SSFML,SSEF,SSEANG,SSDF,SSFEND,SSFND1
     1                ,SSDLE,A4FACT,BRLX,CURRLX,TSIC
       INTEGER        SSFML
       LOGICAL              SSEF,       SSDF,   SSDLE
      COMMON /ERASE2/ AREA(96),AREAO(96),DISP(96),PT(96),LAMBDA(96),
     1                RHO(96),SQRTVV(96),TS(96),TT(96),VMSQ(96),
     2                VVKQKP(96),
     2                WQA(96),WSTA(96), RG(96),C2CP(96),FGR(96)
       REAL           LAMBDA
       DIMENSION      ES2(96),SDNQRM(96)
       EQUIVALENCE    (ES2,VVKQKP),(SDNQRM,RHO)
       DIMENSION      RCU(96)
       EQUIVALENCE    (RCU,LAMBDA)
C      INDEX- M=MO,NM
      COMMON /CZ     / Z(300)
      COMMON /CR     / R(300)
      COMMON /CS2    / S2(300)
      COMMON /CS1    / S1(300)
      COMMON /CPHI1  / PHI1(300)
      COMMON /CM     / JMS(300)
      COMMON /CCURV  / CURV(300)

      COMMON /CB     / B(300)
      COMMON /CIDEX  / M,J,MU,MD,ISTAG
      COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
     *                LO,LESTA,  LOUM(8),
     *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
     *                LEO,LEE, LRO,LRE,LRD
       DIMENSION      LIMITS(24)
       EQUIVALENCE    (LIMITS,LHO)
      COMMON /SLTAB  / W(128),X2(128),SLCHN(128)
       INTEGER SLCHN
      COMMON /CHDATA/ CHNAM(1),LHNEXT(1),WTFLOW(1),TTO(1),PTO(1),
     *                TSO(1),PSO(1),MACHO(1),AO(1),VARY(5),TAB(6)
       INTEGER        CHNAM
       LOGICAL                          VARY
C     FLOW ADJUSTMENT TABLE
C      INDEX- LF=LFO,LFE
C      NFCOLS= 8
C      X1F    = ORTHOGONAL COORDINATE
C      X2F    = STREAMLINE COORDINATE OF SL EMINATING FROM T.E.
C      X1BF   = X1-COORDINATE OF CHOKE STATION OF FLOW BELOW T.E.
C      X1AF   = X1-COORDINATE OF CHOKE STATION OF FLOW ABOVE T.E.
C      S1F    = S1-COORDINATE OF T.E. (UPPER SURFACE). THIS ITEM
C               IS USED WHEN INTERPOLATING FOR WAKE DELTA-STAR.
```

```
C      LFB,LFA=INDICES OF STATIONS BELOW AND ABOVE T.E.
C      NCHB,NCHA=NUMBER OF CHANNELS BELOW AND ABOVE T.E.
C      LRF    = INDEX OF DUMMY ORICHN LIST FOR THE T.E.
C      LRXF   = INDEX OF LAST CHANNEL BELOW THE T.E.
C      JORDER= 0 IF TOTAL FLOW AT X1F IS GIVEN
C            = 2 IF FLOW ABOVE T.E. IS GIVEN
C            = 1 IF FLOW BELOW T.E. IS GIVEN
C      JORDER= -1 IF FLOW AT X1F IS CHOKED AND SINGLE CHANNEL
       DIMENSION       X1F(1),X2F(1),X1BF(1),X1AF(1),
      1                S1F(1),NCHB(1),NCHA(1),JORDER(1),VNR(12)
        EQUIVALENCE    (LFB,X1BF),(LFA,X1AF),(LRF,NCHB),(LRXF,NCHA)
        DIMENSION      LFB(1),LFA(1),LRF(1),LRXF(1)
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
       DIMENSION       X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
      1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
      1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
      3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
        LOGICAL        PRIM
        INTEGER TYPELB,TYPEUB
        DIMENSION      SCHOKE(1)
        EQUIVALENCE    (SCHOKE,DWDV)
       EQUIVALENCE    (CHNAM,X1F,X1), (LHNEXT,X2F,LNEXT)
       EQUIVALENCE    (WTFLOW,X1BF,MLB), (TTO,X1AF,MUB), (PTO,S1F,PRIM)
       EQUIVALENCE    (TSO,NCHB,TYPELB), (PSO,NCHA,NAMELB)
       EQUIVALENCE    (MACHO,JORDER,ILB), (AO,VNR,FLB), (VARY(1),S1LB)
       EQUIVALENCE    (VARY(2),TYPEUB), (VARY(3),NAMEUB), (VARY(4),IUB)
       EQUIVALENCE    (VARY(5),FUB)
        EQUIVALENCE    (TAB(1),AREATB,S1UB), (TAB(2),VMB), (TAB(3),DWDV)
        EQUIVALENCE    (TAB(4),X2CL), (TAB(5),VCL), (TAB(6),MCL)

       COMMON /CINNER/ INRCTR,RDUM,NINNER(16),CNVF(16)
       COMMON /CPRINT/ CDUM(6),PDUM(10)
       COMMON /CQIREM/ YTOL,YO,DYDX,CTRMAX
       COMMON /CTABPR/ I1TAB

C      BEGIN LOOP THROUGH FLOW ADJUSTMENT TABLE
       LF     = LFO
  101  IF(LF.GE.LFE) GO TO 300
       PLB    = 0.
       PUB    = 0.
       WF     = 0.
       CHOKE  = .FALSE.
       SUBSON= .TRUE.
       X1TE   = X1F(LF)
       X2TE   = X2F(LF)
       LXA    = 1
       IF(JORDER(LF).LT.0) GO TO 118

C      SEARCH FOR THE TWO STATIONS AT X1F(LF)
       CALL STAX1(X1TE,X2TE,X2TE,LXB,LXA)

C      SEARCH FOR CHOKE STATION IF THE FLOW IS CHOKED UPSTREAM
       LKB    = LXB
       LKA    = LXA
```

```
      IF(SIFTE.LE.0.) GO TO 190
      N      = (LWNEXT(LW)-2)/2
      LSTAR = LW+N
      CALL LSPFIT(S1W(LW),DST(LSTAR),N, SIFTE,DISP(K-1),1, 0)
      IF(DISP(K-1)) 184,184,186
  184 DISP(K-1)=-1.
      GO TO 190
  186 WAKE   = .TRUE.

C     LOOP FOR NEXT CHANNEL
  190 WADD   = WSTA(K-1)
      GO TO 105

C     USE CONSTANT DENSITY APPROXIMATION FOR  MAJCTR.LE.NODENS
  200 IF(MAJCTR.LE.NODENS) CALL SETM(1,0.,FGRX,K-1)
      RETURN
      END
      OVERLAY(STC,2,1)
```

```
C      X2F    = STREAMLINE COORDINATE OF SL EMINATING FROM T.E.
C      X1BF   = X1-COORDINATE OF CHOKE STATION OF FLOW BELOW T.E.
C      X1AF   = X1-COORDINATE OF CHOKE STATION OF FLOW ABOVE T.E.
C      S1F    = S1-COORDINATE OF T.E. (UPPER SURFACE). THIS ITEM
C               IS USED WHEN INTERPOLATING FOR WAKE DELTA-STAR.
C      LFB,LFA=INDICES OF STATIONS BELOW AND ABOVE T.E.
C      NCHB,NCHA=NUMBER OF CHANNELS BELOW AND ABOVE T.E.
C      LRF    = INDEX OF DUMMY ORTCHN LIST FOR THE T.E.
C      LRXF   = INDEX OF LAST CHANNEL BELOW THE T.E.
C      JORDER= 0 IF TOTAL FLOW AT X1F IS GIVEN
C            = 2 IF FLOW ABOVE T.E. IS GIVEN
C            = 1 IF FLOW BELOW T.E. IS GIVEN
C      JORDER= -1 IF FLOW AT X1F IS CHOKED AND SINGLE CHANNEL
       DIMENSION        X1F(1),X2F(1),X1BF(1),X1AF(1),
      1                 S1F(1),NCHB(1),NCHA(1),JORDER(1),VNR(12)
        EQUIVALENCE     (LFB,X1BF),(LFA,X1AF),(LRF,NCHB),(LRXF,NCHA)
        DIMENSION       LFB(1),LFA(1),LRF(1),LRXF(1)
        EQUIVALENCE     (CH,X1F,X2W), (LTNEXT,X2F,LWNEXT), (NPT,X1BF,S1W)
        EQUIVALENCE     (LPSI,X1AF), (LTT,S1F), (LPT,NCHB), (LRCU,NCHA)
        EQUIVALENCE     (CRG,JORDER), (CPGJ,VNR)

       COMMON /SLTAB / W(128),X2(128),SLCHN(128)
        INTEGER SLCHN
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
        DIMENSION       LIMITS(24)
        EQUIVALENCE     (LIMITS,LHO)

       COMMON /CIDEX / M,J,MU,MD,ISTAG
       COMMON /CMAXIT/ MAXIT,MAJCTR,GREFIN,EDUM
       COMMON /CPTMOV/ VELPOT,ICOB,NODENS,CPTDUM
       COMMON /CR    / R(300)
       COMMON /CS1   / S1(300)
       COMMON /CTHICK/ NTHKX,NTHKY,THKX(20),THKY(20),THIK2D(78)
       COMMON /CZ    / Z(300)
       COMMON /ERASE / PSI(800)

        INTEGER         CHX

C     INTERPOLATE FOR LAMINA THICKNESS
       NK     = MB-MA+1
       CALL SETM(1.,1., LAM,NK)
       IF(NTHKX.LE.1) GO TO 100
       CALL LFIT2D(Z(MA),R(MA),LAM,NK)

C     INITIALIZE
  100 WAKE   = .FALSE.

C     DEFINE NUMBER OF STREAMLINES,  NK,  ASSOCIATED WITH EACH CHANNEL
       K      = 1
       M      = MA
       WADD   = 0.
  105 NK     = 0
       K1     = K
       M1     = M
  110 CALL GETIX
```

165

```
      IF(M.NE.M1) GO TO 114
      CHX    = SLCHN(J)
      PSI1   = X2(J)
  114 IF(SLCHN(J).NE.CHX) GO TO 120
      NK     = NK+1
      DISP(K)=0.
      WSTA(K)=W(J)+WADD
      PSI(NK)=X2(J)
      K      = K+1
      M      = M+1
      IF(M.LE.MB) GO TO 110

C     FIND INDEX IN CONVTB
  120 LT     = LTO
  125 IF(LT.GT.LTE) CALL ERROR1
      IF(CH(LT).EQ.CHX) GO TO 130
      LT     = LT+LTNEXT(LT)
      GO TO 125

C     INTERPOLATE FOR CONVECTED PROPERTIES
C       SCALE THE PSI TABLE TO CONFORM TO THE LPSI-TABLE IN /CONVTB/
  130 NI     = NPT(LT)
      I      = LT+LPSI(LT)
      I2     = I+NI
      IF(K1.EQ.1 .AND. NK.EQ.1) PSI1=PSI1-8.
      PSI1   = 8.*AINT(PSI1/8.)
      F      = XCH(I2-1)/8.
      DO 140 KN=1,NK
  140 PSI(KN)=(PSI(KN)-PSI1)*F
      IT     = LT+LTT(LT)
      IP     = LT+LPT(LT)
      IS     = LT+LRCU(LT)
      CALL LSPFIT(CH(I),CH(IT),NI, PSI,TT(K1),NK, 0)
      CALL LSPFIT(CH(I),CH(IP),NI, PSI,PT(K1),NK, 0)
C     CALL LSPFIT(CH(I),CH(IS),NI, PSI,RCU(K1),NK, 0)
      CALL SETM(1,CRG(LT),RGX(K1),NK)
      CALL SETM(1,C2CP(LT),C2CPX(K1),NK)
      CALL SETM(1,FGR(LT),FGRX(K1),NK)

C     WAKE DISPLACEMENT THICKNESS
C       SEARCH FOR X2-SUBTABLE
      IF(M.GT.MB) GO TO 200
      X2J    = X2(J)
      DISP(K-1)=-1.
      LW     = LWO
  155 IF(LW.GE.LWE) GO TO 190
      IF(X2W(LW).EQ.X2J) GO TO 170
      LW     = LW+LWNEXT(LW)
      GO TO 155
C       FIND TRAILING EDGE S1 IN THE FLOW ADJUSTMENT TABLE, S1F
  170 LF     = LFO
  175 IF(X2F(LF).EQ.X2J) GO TO 180
      LF     = LF+NFCOLS
      IF(LF.LT.LFE) GO TO 175
      CALL ERROR1
C       INTERPOLATE FOR WAKE DISPLACEMENT THICKNESS, DSTAR
  180 S1FTE=S1(M)-S1F(LF)
C       S1-FROM-T.E.
```

```
  160 TO(M) = F*T2(M)+(1.-F)*T1(M)

      M      = M+1
      IF(M.LE.NXY) GO TO 100
C...  END LOOP FOR INTERPOLATIONG TO(M) AT X(M),Y(M),M=1,NXY

      RETURN
      END
```

```
*DECK TTPT
      SUBROUTINE TTPT(MA,MB, WSTA,DISP,WAKE,TT,PT,LAM,RGX,C2CPX,FGRX)
*TTPT--          TT, PT, AND RCU FOR STREAMLINES                -TTPT-
       LOGICAL                              WAKE
       REAL                                          LAM(25)
       DIMENSION WSTA(25),DISP(25),TT(25),PT(25),
     1           RGX(25),C2CPX(25),FGRX(25)


C     INPUT-
C      MA     = FIRST FIELD POINT
C      MB     = LAST FIELD POINT

C     OUTPUT-
C      WSTA   = LIST OF STREAM FUNCTION VALUES
C      DISP(K)=NON-ZERO FOR POSSIBLE SLIP CONDITION BETWEEN STREAMLINE
C                K AND K+1,  OTHERWISE DISP(K)=0.
C              = DISPLACEMENT THICKNESS OF WAKE IF POSITIVE
C      WAKE   = .TRUE. IF THERE EXISTS ANY WAKE DISPLACEMENTS.
C      TT     = INTERPOLATED TOTAL TEMPERATURE
C      PT     = INTERPOLATED TOTAL PRESSURE
C      LAMBDA= LAMINA THICKNESS IN THIRD DIMENSION,  BLOCKAGE EFFECT
C      RCU    = INTERPOLATED ANGULAR MOMENTUM          ***NOT NOW IN USE
C      RGX    = GAS CONSTANT
C      C2CPX  = SPECIFIC HEAT
C      FGRX   = 1./(GAM-1.)= FUNCTION OF GAMMA FOR CALCULATING DENSITY
C      NOTE - LENGTH OF WSTA,TT,PT,RCU-LISTS IS MB-MA+1

C     TABLE OF CONVECTED PROPERTIES
C      INDEX- LT=LTO,LTE
C      CH     = CHANNELNAME
C      LTNEXT= INDEX INCREMENT TO THE NEXT CHANNEL
C      LPSI   = RELATIVE LOCATION OF PSI LIST
C      NPT    = NO. OF PSI, TT, PT AND RCU VALUES
C      LTT    = RELATIVE LOCATION OF TT LIST
C      LPT    = RELATIVE LOCATION OF PT LIST
C      LRCU   = RELATIVE LOCATION OF RCU LIST
       COMMON /CHDATA/ CH(1),LTNEXT(1),NPT(1),LPSI(1),LTT(1),LPT(1),
     1                 LRCU(1),
     2                 CRG(1),CPGJ(1),C2CP(1),QGAM(1),FGT(1),FGP(1),
     3                 FGR(1),AREATB(485)
        INTEGER CH
       DIMENSION XCH(1)
       EQUIVALENCE (CH,XCH)
C     TABLE OF WAKE DISPLACEMENT THICKNESS
C      INDEX- LW=LWO,LWE
       DIMENSION              X2W(1),LWNEXT(1),S1W(47)
        DIMENSION      DST(1)
        EQUIVALENCE    (DST,S1W)
C      SUBTABLE ARRANGEMENT IS-
C       X2W,LWNEXT(=2+2N), S1W(1),S1W(2)...S1W(N), DST(1),DST(2),..DST(N)
C      X2W    = STREAMLINE COORDINATE
C      S1W    = DISTANCE ALONG STREAMLINE FROM T.E.
C      DST    = WAKE DISPLACEMENT THICKNESS AS A FUNCTION OF S1W
C     FLOW ADJUSTMENT TABLE
C      INDEX- LF=LFO,LFE
C      NFCOLS= 8
C      X1F    = ORTHOGONAL COORDINATE
```

```
      IF(SQMD.GE.1.) GO TO 920
      DPTR   = 1. - ((6.*SQM)/(SQM+5.))**3.5 * (6./(7.*SQM-1.))**2.5
      PTR(J)= PTR(J) * (1.-PDUM(18)*DPTR)
  920 M      = M+1
      K      = K+1
      IF(M.LE.MB) GO TO 910
      RETURN
 1568 FORMAT(/2X57H***  ERROR IN FLOBAL, REQUESTED BOUNDARY PRESSURE EXC
     *EEDS/6X37HTOTAL PRESSURE AT TRAILING EDGE POINTF11.5,1H,F11.5,1H./
     *6X3HPS=F8.3,3X3HPT=F8.3,)
      END
```

```
*DECK LFIT2D
      SUBROUTINE LFIT2D(X,Y,TO,NXY)
*LFIT2D          LINEAR SURFACE INTERPOLATION                    -LFIT2D-
C                IN A RECTANGULAR GRID
      DIMENSION        X(2),Y(2),TO(2)

C     INPUT-
C      X,Y    = LIST OF COORDINATES AT WHICH INTERPOLATED VALUES ARE TO BE
C      NXY    = NO OF COORDINATE POINTS

C      NXT    = NUMBER OF XT
C      NYT    = NUMBER OF YT
C      XT     = X-GRID OF T-TABLE
C      YT     = Y-GRID OF T-TABLE
C      T      = TABLE OF VALUES
C      NOTE   - NUMBER OF T-VALUES IS NXT*NYT, ORDER IS ILLUSTRATED BELOW
C         YT(NYT)-   T(3)       T(6)       T(NXT*NYT)
C         YT(2)  -   T(2)       T(5)       T(8)
C         YT(1)  -   T(1)       T(4)       T(7)
C                  ------------------------------------
C                     XT(1)     XT(2)      XT(NXT)

C     OUTPUT-
C      TO     = INTERPOLATED VALUES AT X,Y

      COMMON /CTHICK/ NXT,NYT,XT(20),YT(20),T(78)
      COMMON /ERASE / DUM(400),T1(200),T2(200)

C     FIND CORRECT X-INTERVAL
      I      = 1
      M      = 1
      ISV    = 0
  100 NCOUNT= 0
  105 IF(X(M).LT.XT(I)) GO TO 110
      IF(X(M).GT.XT(I+1)) GO TO 120
      F      = (X(M)-XT(I))/(XT(I+1)-XT(I))
      GO TO 150
  110 IF(I.EQ.1) GO TO 140
      I      = I-1
      GO TO 125
  120 IF((I+1).GE.NXT) GO TO 145
      I      = I+1
  125 NCOUNT= NCOUNT+1
      IF(NCOUNT.GT.NXT) CALL ERROR1
      GO TO 105
  140 F      = 0.
      GO TO 150
  145 F      = 1.

C     INTERPOLATE WRT Y
  150 IF(I.EQ.ISV) GO TO 160
      IJ2    = I*NYT+1
      IJ1    = IJ2-NYT
      CALL LFIT1(YT,T(IJ1),NYT, Y,T1,NXY)
      CALL LFIT1(YT,T(IJ2),NYT, Y,T2,NXY)
      ISV    = I

C     INTERPOLATE WRT X
```

170

```
  630 VMSQ(K)= AMAX1(VMSQ(K),1.E-6)
      IF(ABS(VMSQ(K)/VMSQK-1.).GE.2.E-5) GO TO 620
      GO TO 610
C.....END INTEGRATION OF MOMENTUM EQUATION

C**   INTEGRATION OF FLOW AREA
  650 AREA(1)=AREAO(1)
      M      = MA
      DO 660 K=1,NK
      VM(M) = SQRT(VMSQ(K))
      IF(TS(K).LT.0. .AND. FGR(K).NE.0.) GO TO 590
      RHO(K)= PT(K)/(RG(K)*TT(K)) * (TS(K)/TT(K))**FGR(K)
      WQA(K)= RHO(K)*VM(M)
      IF(M.EQ.MA) GO TO 660
C     NOTE - AVERAGE FLOW/AREA IS APPROXIMATELY SQRT(WQA(K-1)*WQA(K))
      WQAVG = WQA(K)+WQA(K-1)
      X      = (WQA(K)-WQA(K-1))*(WQA(K)-WQA(K-1))/(WQAVG*WQAVG)
      AREA(K)=AREA(K-1) + 2.*(WSTA(K)-WSTA(K-1)) /
     1                        (WQAVG*(1.-X*(.5+X*(.125+X*.0625))))
      IF(DISP(K-1).LE.0.) GO TO 660
      PERIM = 1.
      IF(.NOT.AXIA) GO TO 655
      PERIM = PI*(R(M)+R(M-1))
  655 AREA(K)=AREA(K-1)+DISP(K-1)*PERIM
  660 M      = M+1
C...  END FLOW AREA INTEGRATION

C     RECIPROCAL OF CALCULATED FLOW AREA, ETC.
      QAREA = 1./AREA(NK)
      VMBC   = VM(MB)
      IF(PLB.LT.0. .OR. PUB.NE.0.) GO TO 740
      VMSQSV= VMSQ(NK)
      VVSAFE= VMSQSV
      IF(VMLBSQ.NE.0.) GO TO 710

C     CALL -QIREM- FOR THE NEXT QUESS OF  VM(NK)=VMBC
      IF(QV(1).NE.0.) GO TO 680
      YO     = 1./TAREA
      YTOL   = 1.E-5*YO
      IF(WF.NE.0.) YO=YO*WF/WSTA(NK)
      DYDX   = DWDV(L)
      IF(DYDX.EQ.0. .OR. DYDX.EQ.XCHOKE) DYDX=YO/VMBC
      IF(.NOT.CHOKE) GO TO 675
      YO     = YO+YO
  675 QAREA1= QAREA
      VUB1   = VMBC
  680 XJP    = -.75*VMBC
      IF(.NOT.SUBSON) XJP=.25*VMBC
      CALL QIREM(VMBC,QAREA, XJP,QV)
      IF(QV(1).EQ.0.) GO TO 682
      IF(QV(5).EQ.0.) GO TO 684
      VMSQ(NK)=VMBC*VMBC
      GO TO 600

C     EVALUATE  D(W)/D(VLB),  SAVE VELOCITIES
  682 BOT    = VMBC-VUB1
C     .001 FOR CDC VERSION ONLY
      IF(ABS(BOT).GT..001) DWDV(L)=(QAREA-QAREA1)/BOT
```

```
      GO TO 740

C     THE FLOW IS CHOKED
  684 IF(CHOKE) GO TO 740
      RATIO = QAREA*TAREA
      DO 686 K=1,NK
  686 AREA(K)=RATIO*AREA(K)
      CALL ADJWF2
      GO TO 740

C     CALL -QIREM- FOR LOWER BOUNDARY PRESSURE ITERATION
  710 YO    = VMLBSQ
      YTOL  = 1.E-5*YO
      DYDX  = 1.
      CALL QIREM(VMSQ(NK),VMSQ(1),-.5*VMSQ(NK),QV)
      IF(QV(1).NE.0.) GO TO 600

C     CONTROL STREAMLINE VELOCITY
  715 CONTINUE

C     CALCULATE BOUNDARY PRESSURE
  740 PLBC  = RHO(1)*RG(1)*TS(1)
      PUBC  = RHO(NK)*RG(NK)*TS(NK)
      WRQST = WSTA(NK)
      WCALC = WRQST*QAREA*TAREA

      IF(PDUM(9).LE.0.) GO TO 900
      IF(X1(L).GE.PDUM(8) .AND. X1(L).LE.PDUM(9)) GO TO 800
      GO TO 900
  800 CALL TABPRT(3HSTA,X1(L),1,1)
      CALL TABPRT(6HFB-CFB,L,33,4)
      I1TAB = MA
      CALL TABPRT(5HFB-S2,S2,MB,10)
      I1TAB = MA
      CALL TABPRT(4HPHI1,PHI1,MB,10)
      I1TAB = MA
      CALL TABPRT(4HCURV,CURV,MB,10)
      I1TAB =MA
      CALL TABPRT(5HFB-VM,VM,MB,10)
      CALL TABPRT(6HERASE2,AREA,1536,8)
      CALL TABPRT(4HFB-W,W,NJ,10)
      CALL TABPRT(5HFB-X2,X2,NJ,10)
      CALL TABPRT(5HSLCHN,SLCHN,NJ,10)

C     RESET PLB AND PUB INDICATORS
  900 PLB   = 0.
      PUB   = 0.

C     COMPUTE SHOCK LOSS
      IF(PDUM(18).EQ.0.) RETURN
      K     = 1
      M     = MA
  910 SOM   = VMSQ(K)/(1.4*RG(K)*TS(K))
      IF(SOM.LE.1.) GO TO 920
      CALL GETIX
      IF(MD.EQ.0) GO TO 920
      VVMXSQ= VM(MD)*VM(MD)/(C2CP(K)*TT(K))
      SOMD  = 5.*VVMXSQ/(1.-VVMXSQ)
```

```
      CFGT   = 1./(1.+FGR(1))
      VMLBSQ= C2CP(1)*TT(1)*(1.-(PLB/PT(1))**CFGT)
      GO TO 540
C      FREE OR FIELD LOWER BOUNDARY
  532 IF(TYPELB(L).NE.FREE .AND. TYPELB(L).NE.FIELD) GO TO 534
      M      = MA
      CALL GETIX
      IF(MU.EQ.0) CALL ERROR1
      VMLBSQ= VM(MU)*VM(MU)
  533 PLB    = 1.E-6
      IF(TYPELB(L).NE.FIELD .OR. PASS1) GO TO 540
      VMUBSQ= 0.
      IF(TYPEUB(L).EQ.FIELD) GO TO 570
C      STREAMWISE INTERPOLATION OF VELOCITY AT ISTAG=3 POINT BY  LSPFIT
      IRET   = 1
 5331 M4     = M
      CALL GETRLX
      II     = 0
      NII    = 3
      IF(M2.EQ.M4) GO TO 5333
      II     = 1
      NII    = 4
      S1B(II)=S1(M2)
      V1B(II)=VM(M2)
 5333 S1B(II+1)=S1(M3)
      V1B(II+1)=VM(M3)
      S1B(II+2)=S1(M5)
      V1B(II+2)=VM(M5)
      S1B(II+3)=S1(M6)
      V1B(II+3)=VM(M6)
      IF(M6.EQ.M4) NII=NII-1
      CALL LSPFIT(S1B,V1B,NII, S1(M),VMM,1, 0)
      IF(IRET) 5335,5435,5335
 5335 VMLBSQ= VMM*VMM
      GO TO 540
C      FAR-FIELD LOWER BOUNDARY
  534 IF(TYPELB(L).NE.FARFLD) GO TO 540
      CALL ERROR1
      CALL LSPFIT(ZDN,UDN,25, Z(MA),VMLBSQ,1, 0)
      VMLBSQ= VMLBSQ*VMLBSQ
      GO TO 533

C    UPPER BOUNDARY
  540 VMUBSQ= 0.
C      PRESSURE UPPER BOUNDARY
      IF(PUB.GT.0.) GO TO 541
      IF(TYPEUB(L).NE.PRES) GO TO 542
      CALL LSPFIT(ZP,PPS,NZP, Z(MB),PUB,1, 0)
  541 PSB    = PUB
      PTB    = PT(NK)
      M      = MB
      IF(PSB.GE.PTB) GO TO 568
      CFGT   = 1./(1.+FGR(NK))
      VMUBSQ= C2CP(NK)*TT(NK)*(1.-(PUB/PT(NK))**CFGT)
      GO TO 570
C      FREE OR FIELD UPPER BOUNDARY
  542 IF(TYPEUB(L).NE.FREE  .AND. TYPEUB(L).NE.FIELD) GO TO 544
      M      = MB
```

```
      CALL GETIX
      IF(MU.EQ.0) CALL ERROR1
      VMUBSQ= VM(MU)*VM(MU)
      VMLBSQ= 0.
  543 PUB   = 1.E-6
      IF(TYPEUB(L).NE.FIELD .OR. PASS1) GO TO 570
      IRET  = 0
      GO TO 5331
 5435 VMUBSQ= VMM*VMM
      GO TO 570
C      FAR-FIELD UPPER BOUNDARY
  544 IF(TYPEUB(L).NE.FARFLD) GO TO 570
      CALL LSPFIT(ZDN,UDN,25, Z(MB),VMUBSQ,1, 0)
      VMUBSQ= VMUBSQ*VMUBSQ
      GO TO 543
  568 WRITE (6,1568) Z(M),R(M),PSB,PTB
      CALL ERROR1

C     BEGIN FLOW BALANCE ITERATION
  570 QV(1) = 0.
      IF(VMUBSQ.NE.0.) VMSQ(NK)=VMUBSQ
      VMSQSV= VMSQ(NK)

C     NEGTS,VVSAFE ARE USED FOR SALVAGING NEGATIVE TEMPERATURE SITUATIONS
      NEGTS = 0
      VVSAFE= 0.
      GO TO 600
  590 NEGTS = NEGTS+1
      IF(NEGTS.GE.20 .OR. (PLB+PUB).NE.0.) CALL ERROR1
      VMSQ(NK)=.5*(VMSQ(NK)+VVSAFE)

C****STEP BY STEP INTEGRATION OF NORMAL MOMENTUM EQUATION
  600 VRATIO= VMSQ(NK)/VMSQSV
      K     = NK

C     PREDICT VELOCITY AT K
  610 K     = K-1
      IF(K) 615,650,615

C     COEFFICIENT VALUES AT K+1
  615 TS(K+1)=TT(K+1)-VMSQ(K+1)/C2CP(K+1)
      CDPT1 = RG(K+1)*TS(K+1)/PT(K+1)

C     COEFFICIENT VALUES AT K
      VMSQ(K)=VMSQ(K)*VRATIO
  620 VMSQK = VMSQ(K)
      TS(K) = TT(K)-VMSQ(K)/C2CP(K)
      CDPT  = CDPT1 + RG(K)*TS(K)/PT(K)

C     INTEGRATE
      IF(DISP(K).NE.0.) GO TO 625
  622 VMSQ(K)=VMSQ(K+1)*VVKQKP(K) + SQRTVV(K)*(CDPT*(PT(K)-PT(K+1)))
      GO TO 630
C     (WAKE DISCONTINUITY)
  625 IF(PT(K+1).EQ.PT(K)) GO TO 622
      PSLIP = PT(K+1)*(TS(K+1)/TT(K+1))**(FGR(K+1)+1.)
      TS(K) = TT(K)*(PSLIP/PT(K))**(1./(1.+FGR(K)))
      VMSQ(K)=C2CP(K)*(TT(K)-TS(K))
```

```
C       INDEX- M=MO,NM
        COMMON /CZ     / Z(300)
        COMMON /CR     / R(300)
        COMMON /CS2    / S2(300)
        COMMON /CS1    / S1(300)
        COMMON /CPHI1  / PHI1(300)
        COMMON /CM     / JMS(300)
        COMMON /CCURV  / CURV(300)
        COMMON /CB     / B(300)
        COMMON /CVM    / VM(300)
        COMMON /CIDEX  / M,J,MU,MD,ISTAG
        COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
       *                LO,LESTA,  LDUM(8),
       *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
       *                LEO,LEE, LRO,LRE,LRD
         DIMENSION      LIMITS(24)
         EQUIVALENCE    (LIMITS,LHO)
        COMMON /CBEND  / NBCB(2),ANGE(2),CURVE(2),FB(2)
        COMMON /CBITS  / BITS,BLANK
        COMMON /CCUBE  / NBC(2),C1(2),C2(2),FEND(2)
        COMMON /CGRAV  / CG
        COMMON /CPI    / PI,TWOPI,PIQ2,PIQ4,TODEG,TORAD
        COMMON /CREFIN/ SG1,SG2,VMG1,VMG2
       1,               NGR,NGZ, SGR(10),GR(10), SGZ(10),GZ(10)
        COMMON /SLTAB  / W(128),X2(128),SLCHN(128)
         INTEGER SLCHN
        COMMON /SLTAB2/ PTR(128)
C      STATION TABLE
C       INDEX- L=LO,LESTA
C       SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C       MCL    = SHARP CORNER INDICATOR (BLDTBS)
C       MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
        COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
       1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
       1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
       3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
         LOGICAL        PRIM
         INTEGER TYPELB,TYPEUB
         DIMENSION      SCHOKE(1)
         EQUIVALENCE    (SCHOKE,DWDV)

        COMMON /CFB2   / PASS1
         LOGICAL        PASS1
        COMMON /CFRFIN/ ATINF,MINF,RFFINF,UINF,ZDN1,ZDN25
        COMMON /CFRFLD/ NFF(130),ZDN(50),UDN(25)
        COMMON /CIDEXR/ M4,CI1(4),M3,CI2(4),M5,CI3(4),M2,CI4(4),M6,CI5(4)
        COMMON /CISBOT/ CISDUM(7), NZP,ZP(10),PPS(10)
        COMMON /CLSPF  / I,LEND
         LOGICAL          LEND
        COMMON /CPRINT/ PRTES2(6),PDUM(20)
        COMMON /CQIREM/ YTOL,YO,DYDX,CTRMAX
        COMMON /CTABPR/ ILTAB

        INTEGER         FARFLD,FREE,PRES,FIELD
        LOGICAL         WAKE

        DATA FARFLD/6HFARFLD/, FREE/4HFREE/, PRES/4HPRES/, FIELD/5HFIELD/
```

```
  500 MA     = MLB(L)
      MB     = MUB(L)
      IF(L.EQ.LO) CALL SETM(1,1., PIR,NJ)

C     CHECK FOR STAGNATION BOUNDARY POINT
      M      = MA
      CALL GETIX
      IF(ISTAG.NE.1) GO TO 515
      MA     = MA+1
  515 M      = MB
      CALL GETIX
      IF(ISTAG.NE.1) GO TO 520
      MB     = MB-1

C     BUILD TABLE OF FLOW FUNCTION AND STAGNATION CONDITIONS
  520 CALL ITPT(MA,MB, WSTA,DISP,WAKE,TT,PT,LAMBDA, RG,C2CP,EGR)

C     PASSAGE AREA AND SHOCK PRESSURE LOSS
      K      = 1
      M      = MA
  522 RLAMDA(K)=LAMBDA(K)
      IF(AXIA) RLAMDA(K)=TWOPI*R(M)*LAMBDA(K)
      CALL GETIX
      PT(K) = PT(K)*PTR(J)
      K      = K+1
      M      = M+1
      IF(M.LE.MB) GO TO 522
      AREAO(1)=0.
      NK     = MB-MA+1
      LEND   = .FALSE.
      IF(NK.GT.2 .AND. (DISP(2).NE.0. .OR. DISP(NK-2).NE.0.))LEND=.TRUE.
      CALL LSPFIT(S2(MA),RLAMDA,NK, S2(MA),AREAO,NK, -1)
      TAREA = AREAO(NK)

C     INTEGRATE CURVATURE WITH RESPECT TO S2
C     INITIAL ESTIMATE OF MERIDIONAL VELOCITY SQUARED
      SDNORM= 0.
      CALL LSPFIT(S2(MA),CURV(MA),NK, S2(MA),SDNORM,NK,-1)
      LEND   = .FALSE.
      M      = MA+1
      DO 525 K=2,NK
      VVKOKP(K-1)=EXP(2.*(SDNORM(K)-SDNORM(K-1))) * TT(K-1)/TT(K)
      SQRTVV(K-1)=SQRT(VVKOKP(K-1))
      VMSQ(K-1)=VM(M-1)*VM(M-1)
  525 M      = M+1
      VMSQ(NK)=VMB(L)*VMB(L)

C     SPECIFIED STATIC PRESSURE AND SPECIAL BOUNDARY OPTIONS
      VMLBSQ= 0.
C       PRESSURE LOWER BOUNDARY
      IF(PLB.GT.0.) GO TO 530
      IF(TYPELB(L).NE.PRES) GO TO 532
      CALL LSPFIT(ZP,PPS,NZP, Z(MA),PLB,1, 0)
  530 PSB    = PLB
      PTB    = PT(1)
      M      = MA
      IF(PSB.GE.PTB) GO TO 568
```

```
C       (BELOW THE BODY)
  125 IF(X2(JB).EQ.X2F(LF)) X1BF(LF)=X1(L)
C       (ABOVE THE BODY)
      IF(X2(JA).EQ.X2F(LF)) X1AF(LF)=X1(L)
      RETURN

C     CHOKED CHANNEL W/O T.E., ADD A LINE TO /CADJWF/
  200 LF     = LFE+1
      IF(LF.NE.LO) GO TO 205
      NMOVE  = LO-LESTA-1
      LO     = LO+NFCOLS
      CALL MOVE(1, X1(LF),X1(LO),NMOVE,1)
      CALL SETM(1,0, X1F(LF),NFCOLS)
      L      = L+NFCOLS
      LSTE   = LSTE+NFCOLS
      LESTA  = LESTA+NFCOLS
      LFE    = LFE+NFCOLS
  205 X1F(LF)=X1(LSTE)
      X2F(LF)=X2(JA)
      X1AF(LF)=X1(L)
      X1BF(LF)=X1F(LF)
      JORDER(LF)=-1

C     WRITE COMMENT
  800 WRITE (6,1800) X1(L),L
 1800 FORMAT(/1X32HUNEXPECTED CHOKE,  STATION(X11)=F6.3,4X2HL=I4,)
      IF(LSTE.EQ.0) CALL ERROR1
      RETURN
      END
```

```
*DECK FLOBAL
      SUBROUTINE FLOBAL
*FLOBAL          FLOW BALANCE ROUTINE                              -FLOBAL-

C      INTEGRATION OF THE CONTINUITY AND NORMAL MOMENTUM EQUATIONS
C      ALONG THE ORTHOGONALS TO THE STREAMLINES

C     INPUT-
C      L      = INDEX IN THE STATION TABLE
C      PLB    = LOWER BOUNDARY STATIC PRESSURE IF KNOWN.
C      PUB    = UPPER BOUNDARY STATIC PRESSURE IF KNOWN.
C               EITHER PLB OR PUB OR BOTH MUST BE ZERO.
C               IF PLB (OR PUB) =-1, NO ITERATION FOR FLOW OR
C               PRESSURE IS PERFORMED.
C      WF     = FLOW RATE IF KNOWN (OVERRIDES VALUE OF WSTA)
C      CHOKE  = T FOR CALCULATION OF MAX FLOW

C      S2(M) = DISTANCE ALONG THE ORTHOGONOL
C      CURV(M)=STREAMLINE CURVATURE
C      STATION TABLE
C      VMB(L)= ESTIMATED VELOCITY ON THE UPPER BOUNDARY
C      DWDV(L)=DERIVATIVE OF THE AREA INVERSE WITH RESPECT TO BOUNDARY VE
C      STREAMLINE TABLE

C     OUTPUT-
C      PLBC   = CALCULATED LOWER BOUNDARY PRESSURE, M=MA
C      PUBC   = CALCULATED UPPER BOUNDARY PRESSURE, M=MB
C      TAREA  = TOTAL PASSAGE AREA FOR ALL STREAMTUBES
C      WCALC  = CALCULATED FLOW
C      WRQST  = REQUESTED FLOW (SLTAB DATA)
C      VMBC   = CALCULATED VELOCITY ON THE UPPER BONNDARY
C      DWDV(L)=DERIVATIVE OF THE AREA INVERSE WITH RESPECT TO BOUNDARY VE
C      VCL(L)= VELOCITY ON THE CONTROL STREAMLINE
C      PLB,PUB=0.  (RESET FOR NEXT ENTRY)

       COMMON /ALLCOM/ MACHA,PSA,TSA,PTA,TTA, AXIA,RGA,GAMA,
      1                MACHC,PSC,TSC,PTC,TTC, AXIC,RGC,GAMC,
      2                DAXIT,SCALEA,TTE,CHOTST
         REAL          MACHA(1),MACHC
         LOGICAL       AXIA,AXIC
         LOGICAL       CHOTST
       COMMON /CFB    / L,MA,MB,PLB,PUB,WF,CHOKE,SUBSON, NK,PLBC,PUBC,
      1                XCHOKE, TAREA,VMBC, WRQST,WCALC, QV(8),QVP(8),
      *                JSUM,VMLBSQ
         LOGICAL                                CHOKE,SUBSON
         DIMENSION     S1B(4),V1B(4)
         EQUIVALENCE   (S1B,QV),(V1B,QV(5))
       COMMON /ERASE2/ AREA(96),AREA0(96),DISP(96),PT(96),LAMBDA(96),
      1                RHO(96),SQRTVV(96),TS(96),TT(96),VMSQ(96),
      2                VVKQKP(96),
      2                WQA(96),WSTA(96), RG(96),C2CP(96),FGR(96)
         REAL          LAMBDA
         DIMENSION     ES2(96),SDNORM(96)
         EQUIVALENCE   (ES2,VVKQKP),(SDNORM,RHO)
         DIMENSION     RCU(96)
         EQUIVALENCE   (RCU,LAMBDA)
         DIMENSION     RLAMDA(96)
         EQUIVALENCE   (RLAMDA,AREA)
```

```
*DECK ADJWF2
      SUBROUTINE ADJWF2
*ADJWF2          INSERT CHOKE STATION IN FLOW ADJ-TABLE       -ADJWF2-

C     FLOW ADJUSTMENT TABLE
C      INDEX- LF=LFO,LFE
C      NFCOLS= 8
C      X1F    = ORTHOGONAL COORDINATE
C      X2F    = STREAMLINE COORDINATE OF SL EMINATING FROM T.E.
C      X1BF   = X1-COORDINATE OF CHOKE STATION OF FLOW BELOW T.E.
C      X1AF   = X1-COORDINATE OF CHOKE STATION OF FLOW ABOVE T.E.
C      S1F    = S1-COORDINATE OF T.E. (UPPER SURFACE). THIS ITEM
C               IS USED WHEN INTERPOLATING FOR WAKE DELTA-STAR.
C      LFB,LFA=INDICES OF STATIONS BELOW AND ABOVE T.E.
C      NCHB,NCHA=NUMBER OF CHANNELS BELOW AND ABOVE T.E.
C      LRF    = INDEX OF DUMMY ORTCHN LIST FOR THE T.E.
C      LRXF   = INDEX OF LAST CHANNEL BELOW THE T.E.
C      JORDER= 0 IF TOTAL FLOW AT X1F IS GIVEN
C            = 2 IF FLOW ABOVE T.E. IS GIVEN
C            = 1 IF FLOW BELOW T.E. IS GIVEN
C      JORDER= -1 IF FLOW AT X1F IS CHOKED AND SINGLE CHANNEL
       COMMON /CHDATA/ X1F(1),X2F(1),X1BF(1),X1AF(1),
      1                S1F(1),NCHB(1),NCHA(1),JORDER(1),VNR(12)
        EQUIVALENCE    (LFB,X1BF),(LFA,X1AF),(LRF,NCHB),(LRXF,NCHA)
        DIMENSION      LFB(1),LFA(1),LRF(1),LRXF(1)
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
       DIMENSION       X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
      1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
      1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
      3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
        LOGICAL        PRIM
        INTEGER TYPELB,TYPEUB
        DIMENSION      SCHOKE(1)
        EQUIVALENCE    (SCHOKE,DWDV)
       DIMENSION      TAB(6)
        EQUIVALENCE   (CHNAM,BDT,CH,X2W,X1F,X1)
        EQUIVALENCE   (LHNEXT,LBNEXT,LTNEXT,LWNEXT,X2F,LNEXT)
        EQUIVALENCE   (WTFLOW,LBZ1,NPT,S1W,X1BF,MLB)
        EQUIVALENCE   (TTO,CHNAME,LPSI,X1AF,MUB), (PTO,UP,LTT,S1F,PRIM)
        EQUIVALENCE   (TSO,LEDEX,LPT,NCHB,TYPELB)
        EQUIVALENCE   (PSO,ZBT,LRCU,NCHA,NAMELB)
       EQUIVALENCE   (MACHO,RBT,CRG,JORDER,ILB), (AO,ANGBT,CPGJ,VNR,FLB)
        EQUIVALENCE   (VARY,C2CP,S1LB), (RG,QGAM,TYPEUB)
        EQUIVALENCE   (GAM,FGT,NAMEUB), (NR,FGP,IUB), (NC,FGR,FUB)
        EQUIVALENCE   (TAB(1),AREATB,S1UB), (TAB(2),VMB), (TAB(3),DWDV)
        EQUIVALENCE   (TAB(4),X2CL), (TAB(5),VCL), (TAB(6),MCL)
       COMMON /CFB   / L,MA,MB,PLB,PUB,WF,CHOKE,SUBSON, NK,PLBC,PUBC,
      1                XCHOKE, TAREA,VMBC, WRQST,WCALC, QV(8),QVP(8),
      *                JSUM,VMLBSQ
        LOGICAL                              CHOKE,SUBSON
       COMMON /CSS   / SSFML,SSEF,SSEANG,SSDF,SSFEND,SSFND1
      1               ,SSDLE,A4FACT,BRLX,CURRLX
        INTEGER        SSFML
        LOGICAL              SSEF,       SSDF,   SSDLE
```

```
C      SSFML  = SUPERSONIC CURVATURE FORMULA NUMBER
C      SSEF   = SUPERSONIC ENTERING FLOW, T OR F
C      SSEANG= ENTERING FLOW ANGLE (DEGREES) FOR SSEF=T
C      SSDF   = SUPERSONIC DISCHARGE FLOW, T OR F
C      SSFEND= SUPERSONIC BEAM DOWNSTREAM END CONDITION, =0,1 FOR PARABOL
C      SSFND1= SUPERSONIC BEAM UPSTREAM END CONDITION, =0,1, FOR PARABOLA
C      SSDLE  = SS FLOW BELOW AND AFT OF LE PT, T OR F
C      A4FACT= CENTRAL POINT INFLUENCE COEFFICIENT FACTOR
C      BRLX   = B-RELAXATION FACTOR
C      CURRLX= CURVATURE RELAXATION FACTOR
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
        DIMENSION      LIMITS(24)
        EQUIVALENCE    (LIMITS,LHO)
       COMMON /SLTAB / W(128),X2(128),SLCHN(128)
        INTEGER SLCHN

       COMMON /CIDEX / M,J,MU,MD,ISTAG

C     CHECK FOR SMALLER PREVIOUSLY DECTECTED AREA
       M      = MLB(L)
       CALL GETIX
       JA     = J
       M      = MUB(L)
       CALL GETIX
       JB     = J
       JSUML = JA+256*JB
       IF(JSUM.NE.JSUML) GO TO 90
       IF(TAREA.GT.SVAREA) RETURN
    90 JSUM   = JSUML
       SVAREA= TAREA
       IF(SSDF) SUBSON=.FALSE.

C     SEARCH FORWARD TO TRAILING EDGE
       LX     = L
       LSTE   = 0
   105 IF(.NOT.PRIM(LX)) GO TO 110
       M      = MLB(LX)
       CALL GETIX
       IF(J.NE.JA) GO TO 115
       M      = MUB(LX)
       CALL GETIX
       IF(J.NE.JB) GO TO 115
       LSTE   = LX
   110 LX     = LX+LNEXT(LX)
       IF(LX.LT.LESTA) GO TO 105
   115 IF(LSTE.EQ.0) GO TO 800

C     SEACH CADJWF-TABLE FOR T.E. VALUE OF X1
       LF     = LFO
   120 IF(LF.GE.LFE) GO TO 200
       IF(XIF(LF).EQ.XI(LSTE)) GO TO 125
       LF     = LF+NFCOLS
       GO TO 120

C     IS THE L-ORTHOGONAL BELOW OR ABOVE THE BODY
```

```
      DIMENSION       BDT(1),LBNEXT(1),LBZ1(1),
     1                CHNAME(1),UP(1),LEDEX(1),
     2                ZBT(1),RBT(1),ANGBT(42)
        LOGICAL       UP
        INTEGER BDT,CHNAME,BDNAME
        DIMENSION     BDNAME(1),LBA(1),LBB(1)
        EQUIVALENCE   (BDNAME,ZBT), (LBA,RBT), (LBB,ANGBT)
C     FLOW ADJUSTMENT TABLE
C      INDEX- LF=LFO,LFE
C      NFCOLS= 8
C      X1F    = ORTHOGONAL COORDINATE
C      X2F    = STREAMLINE COORDINATE OF SL EMINATING FROM T.E.
C      X1BF   = X1-COORDINATE OF CHOKE STATION OF FLOW BELOW T.E.
C      X1AF   = X1-COORDINATE OF CHOKE STATION OF FLOW ABOVE T.E.
C      S1F    = S1-COORDINATE OF T.E. (UPPER SURFACE). THIS ITEM
C               IS USED WHEN INTERPOLATING FOR WAKE DELTA-STAR.
C      LFB,LFA=INDICES OF STATIONS BELOW AND ABOVE T.E.
C      NCHB,NCHA=NUMBER OF CHANNELS BELOW AND ABOVE T.E.
C      LRF    = INDEX OF DUMMY ORTCHN LIST FOR THE T.E.
C      LRXF   = INDEX OF LAST CHANNEL BELOW THE T.E.
C      JORDER= 0 IF TOTAL FLOW AT X1F IS GIVEN
C            = 2 IF FLOW ABOVE T.E. IS GIVEN
C            = 1 IF FLOW BELOW T.E. IS GIVEN
C      JORDER= -1 IF FLOW AT X1F IS CHOKED AND SINGLE CHANNEL
       DIMENSION      X1F(1),X2F(1),X1BF(1),X1AF(1),
     1                S1F(1),NCHB(1),NCHA(1),JORDER(1),VNR(12)
        EQUIVALENCE   (LFB,X1BF),(LFA,X1AF),(LRF,NCHB),(LRXF,NCHA)
        DIMENSION     LFB(1),LFA(1),LRF(1),LRXF(1)
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
       COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
     1                 TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
     1                 TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
     3                 VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
        LOGICAL       PRIM
        DIMENSION     SCHOKE(1)
        EQUIVALENCE   (SCHOKE,DWDV)

        EQUIVALENCE   (BDT,X1F,X1), (LBNEXT,X2F,LNEXT), (LBZ1,X1BF,MLB)
        EQUIVALENCE   (CHNAME,X1AF,MUB), (UP,S1F,PRIM)
        EQUIVALENCE   (LEDEX,NCHB,TYPELB), (ZBT,NCHA,NAMELB)
        EQUIVALENCE   (RBT,JORDER,ILB), (ANGBT,VNR,FLB)

       COMMON /CTABPR/ I1TAB

       CALL TABPRT(6HALLCOM,MACHA,20,8)
       CALL TABPRT(3HCFB,L,33,4)
       CALL TABPRT(5HCIDEX,M,5,5)
       I1TAB = LBDO
       CALL TABPRT(6HBDYTAB,BDT,LBDE,3)
       I1TAB = LFO
       CALL TABPRT(6HCADJWF,X1F,LFE,8)
       I1TAB = LO
       CALL TABPRT(6HSTATAB,X1,LESTA,5)
   150 WRITE (6,1150) (J,X2(J),SLCHN(J),W(J),J=1,NJ)
```

```
      CALL TABPRT(5HERASE,ERASEC,800,5)

      CALL JMSPRT
      CALL TABPRT(2HS1,S1,NM,10)
      CALL TABPRT(2HS2,S2,NM,10)
      CALL TABPRT(1HZ,Z,NM,10)
      CALL TABPRT(1HR,R,NM,10)
      CALL TABPRT(4HPHI1,PHI1,NM,10)
      CALL TABPRT(4HCURV,CURV,NM,10)
      CALL TABPRT(2HVM,VM,NM,10)
      CALL TABPRT(1HB,B,NM,10)
      CALL TABPRT(6HERASE2,AREA,1536,8)

      LSTOP = 5
      GO TO (999,999) , LSTOP
  999 RETURN
 1150 FORMAT(///1X17HSTREAMLINE TABLE-/17X32HJ       X2            SLCHN
     *     W/(I18,F12.6,6X,A6,F12.6,),)
      END
```

```
*DECK CFB
      BLOCK DATA CFBBLK
*CFB---          BLOCK DATA FOR CFB                                    -CFB-
      COMMON /CFB    / L,MA,MB,PLB,PUB,WF,CHOKE,SUBSON, NK,PLBC,PUBC,
     1                 XCHOKE, TAREA,VMBC, WRQST,WCALC, QV(8),QVP(8),
     *                 JSUM,VMLBSQ
       LOGICAL                                 CHOKE,SUBSON
      DATA XCHOKE/5HCHOKE/, JSUM/0/
      END
```

```
*DECK ERRORX
      SUBROUTINE ERROR1
CEDUMPX         EDUMP FOR STC EXECUTE SECTION                    -EDUMPX-

       COMMON /ALLCOM/ MACHA,PSA,TSA,PTA,TTA, AXIA,RGA,GAMA,
      1                MACHC,PSC,TSC,PTC,TTC, AXIC,RGC,GAMC,
      2                DAXIT,SCALEA,TTE,CHOTST
        REAL           MACHA(1),MACHC
        LOGICAL        AXIA,AXIC
        LOGICAL        CHOTST
       COMMON /CFB    / L,MA,MB,PLB,PUB,WF,CHOKE,SUBSON, NK,PLBC,PUBC,
      1                XCHOKE, TAREA,VMBC, WRQST,WCALC, QV(8),QVP(8),
      *                JSUM,VMLBSQ
        LOGICAL                              CHOKE,SUBSON
       COMMON /ERASE / ERASEC(800)
       COMMON /ERASE2/ AREA(96),AREAO(96),DISP(96),PT(96),LAMBDA(96),
      1                RHO(96),SQRTVV(96),TS(96),TT(96),VMSQ(96),
      2                VVKQKP(96),
      2                WQA(96),WSTA(96), RG(96),C2CP(96),FGR(96)
        REAL           LAMBDA
        DIMENSION      ES2(96),SDNQRM(96)
        EQUIVALENCE    (ES2,VVKQKP),(SDNQRM,RHO)
        DIMENSION      RCU(96)
        EQUIVALENCE    (RCU,LAMBDA)
C     FIELD TABLES
C      INDEX- M=MO,NM
       COMMON /CZ     / Z(300)
       COMMON /CR     / R(300)
       COMMON /CS2    / S2(300)
       COMMON /CS1    / S1(300)
       COMMON /CPHI1  / PHI1(300)
       COMMON /CM     / JMS(300)
       COMMON /CCURV  / CURV(300)

       COMMON /CB     / B(300)
       COMMON /CIDEX  / M,J,MU,MD,ISTAG
C     TABLE OF INDEX LIMITS
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
        DIMENSION      LIMITS(24)
        EQUIVALENCE    (LIMITS,LHO)
       COMMON /CVM    / VM(300)

C     STREAMLINE TABLE
       COMMON /SLTAB / W(128),X2(128),SLCHN(128)
        INTEGER SLCHN
C     BOUNDARY TABLE
C      INDEX- LB=LBDO,LBDE
C      LBNEXT= INCREMENT TO NEXT BOUNDARY
C      LBZ1  = INCREMENT TO THE FIRST BOUNDARY POINT (=0 BEFORE COALLATIO
C      CHNAME= CHANNEL WITH WHICH THE BOUNDARY DATA IS ASSOCIATED
C      UP    = T OR F FOR UPPER OR LOWER BOUNDARY
C      LEDEX = RELATIVE INDEX OF L.E. POINT WHEN LOWER AND UPPER SURFACE
C              CONTOURS ARE CONNECTED
C      BDNAME,LBA,LBB=NAME AND INDEX LIMITS OF SPECIFIC BOUNDARY
C                     DATA WHEN BOUNDARIES ARE COALLATED
```

```
*DECK RTCFI
      SUBROUTINE RTCFI(CHT1,LH)
*RTCFI-         RETRIEVE CHANNEL FLOW INPUT                    -RTCFI-

C     INPUT-
C      CHDATA= CHANNEL INPUT DATA TABLE
C      CHT1   = CHANNEL NAME

C     OUTPUT-
C      LH     = INDEX OF CHT1 IN THE CHANNEL DATA TABLE
C             = 0  IF NO CHANNEL DATA WAS FOUND
C      IF THEY EXIST, THE CHDATA-LISTS TT,PT,RCU ARE TRANSFERRED TO THE
C      LISTS OF  TT,PT,RCU.  IF THEY DO NOT EXIST, TT,PT,RCU = BITS.

       INTEGER CHT1
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
        DIMENSION     LIMITS(24)
        EQUIVALENCE   (LIMITS,LHO)

       COMMON /CBITS / BITS,BLANK
       COMMON /CHDATA/ CHNAM(1),LHNEXT(1),WTFLOW(1),TTO,PTO,
      1                TSO(1),PSO(1),MACHO(1),AO(1),VARY(1),
      2                RG(1),GAM(1), NR(1),NC(1),TAB(6),
      4                B(75)
       INTEGER  CHNAM
       COMMON /ERASE/  QV(8),EDUM(72), A(90),V(90),
      1                PSI(90),R(90),TT(90),PT(90),RCU(90),PS(90)
        DIMENSION     Y(90)
        EQUIVALENCE   (Y,R)

       NAMELIST /NLCHN / PSI,R,Y,TT,PT,RCU,PS

C     SEARCH CHDATA FOR CHANNEL=CHT1
       LH     = LHO
   60  IF(LH.GE.LHE) GO TO 65
       IF(CHNAM(LH).EQ.CHT1) GO TO 70
       LH     = LH+LHNEXT(LH)
       GO TO 60

C      NO INPUT TABLE WAS FOUND
   65  LH     = 0
       RETURN

C      AN INPUT TABLE WAS FOUND
   70  CONTINUE

C     PLACE THE TABLE IN COMMON-ERASE
       NCR    = NC(LH)*NR(LH)
       IF(NCR.GT.0) CALL ISORT(TT,PT,RCU, B(LH),NCR)
       RETURN
       END
       OVERLAY(STC,2,0)
```

```
*DECK STCB
      PROGRAM STCB
      COMMON /CHNFPT/ ICHN(10),WTFS(10),WTFA(10),WPTO(10),WTTO(10), IC
      COMMON /SELECT/ LENTRY
      GO TO (10,20), LENTRY
C     NORMAL ENTRY-- STATION LOOP, FLOW BALANCE
   10 CALL OVERLAY(3HSTC,2,1,6HRECALL)
      GO TO 30
   20 CALL OVERLAY(3HSTC,2,2,6HRECALL)
      CALL OVERLAY(3HSTC,2,3,6HRECALL)
   30 RETURN
      END
```

```
*DECK RBCONV
      SUBROUTINE RBCONV
*RBCONV         REBUILD CONVECTED PROPERTIES TABLE          -RBCONV-

C     COLLECT LIST OF CHANNELS FROM /CONVTB/, THEN BUILD A
C     NEW /CONVTB/ FROM CHANNEL DATA TO ACCOUNT FOR INPUT MODIFICATIONS

C    TABLE OF CONVECTED PROPERTIES
C     INDEX- LT=LTO,LTE
C     CH     = CHANNELNAME
C     LTNEXT= INDEX INCREMENT TO THE NEXT CHANNEL
C     LPSI   = RELATIVE LOCATION OF PSI LIST
C     NPT    = NO. OF PSI, TT, PT AND RCU VALUES
C     LTT    = RELATIVE LOCATION OF TT LIST
C     LPT    = RELATIVE LOCATION OF PT LIST
C     LRCU   = RELATIVE LOCATION OF RCU LIST
      COMMON /CHDATA/ CH(1),LTNEXT(1),NPT(1),LPSI(1),LTT(1),LPT(1),
     1                LRCU(1),
     2                CRG(1),CPGJ(1),C2CP(1),QGAM(1),FGT(1),FGP(1),
     3                FGR(1),AREATB(485)
       INTEGER CH
      DIMENSION XCH(1)
      EQUIVALENCE (CH,XCH)
      COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
     *                LO,LESTA,  LDUM(8),
     *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
     *                LEO,LEE, LRO,LRE,LRD
       DIMENSION      LIMITS(24)
      COMMON /SLTAB / W(128),X2(128),SLCHN(128)
       INTEGER SLCHN

      COMMON /CFB2  / PASS1
       LOGICAL        PASS1
      COMMON /ERASE2/ CHT(500),AT(500),FLW(500)
      COMMON /SPACER/ MAXLH,MAXLT,MAXLF,MAXLW

       INTEGER        CHT

C    ACCUMULATE CHANNEL NAMES AND AREAS
      LT     = LTO
      I      = 0
  110 I      = I+1
      CHT(I)= CH(LT)
      AT(I) = AREATB(LT)
      LTI    = LT+LPSI(LT)+NPT(LT)-1
      FLW(I)= XCH(LTI)
      LT     = LT+LTNEXT(LT)
      IF(LT.LT.LTE) GO TO 110
      NI     = I

C    CYCLE THROUGH BCONV ROUTINE
      PASS1 = .FALSE.
      LTE    = LTO-1
      I      = 1
  130 CALL BCONV(CHT(I),LT,AT(I))
C      CHECK FOR CHANGED FLOW RATE
      LT     = LT+LPSI(LT)+NPT(LT)-1
      IF(XCH(LT).EQ.FLW(I)) GO TO 190
```

```
C        UPDATE THE STREAMLINE TABLE FLOW VALUES
C          SEARCH FOR FIRST AND LAST ELEMENTS OF SLCHN(J)=CHT(I)
         DO 140 JA=1,NJ
  140    IF(SLCHN(JA).EQ.CHT(I)) GO TO 150
  150    DO 160 J=JA,NJ
         IF(SLCHN(J).NE.CHT(I)) GO TO 170
  160    JB     = J
C          SCALE THE CUMULATIVE FLOW RATE VALUES
  170    DO 180 J=JA,JB
  180    W(J)   = W(J)/W(JB)*XCH(LT)
C       SET PASS1=T TO JUMP AROUND INTERPOLATION FOR VM IN FLOBAL
C     (TYPE=FIELD)
       PASS1 = .TRUE.
  190  I      = I+1
       IF(I.LE.NI) GO TO 130

       IF(LTE.LT.LWO) GO TO 980
       WRITE (6,1960) LTO,LTE,MAXLT,LWO
       CALL ERROR1

  980  RETURN
 1960  FORMAT(/1X69H***  THE TABLE OF CONVECTED PROPERTIES HAS EXCEEDED A
      *LLOCATED MEMORY./6X4HLTO=I4,3X4HLTE=I4,3X6HMAXLT=I4,3X4HLWO=I4,)
       END
```

```
      BDMSLA= 0.
      IF(UP(LB)) BDMSLA=PI
      GO TO 120
  105 LB2    = LB+LEDEX(LB)-3
      BDMSLA= PI
      IF(CHNAME(LB).EQ.NAMCHN) GO TO 120
      LB1    = LB2+3
      LB2    = LB20
      BDMSLA= 0.
      IF(CHNAME(LB+1).EQ.NAMCHN) GO TO 120
      CALL ERROR1

  120 FGE1   = .FALSE.
      DO 150 LB=LB1,LB2,3
      DZ     = ZBT(LB+3)-ZBT(LB)
      DR     = RBT(LB+3)-RBT(LB)
      SB     = SQRT(DZ*DZ+DR*DR)
      IF(SB.EQ.0.) GO TO 150
      CSB    = DZ/SB
      SNB    = DR/SB
C     AP     = ANGLE OF THE PERPENDICULAR OR ORTHOGONAL
      AP     = .50*APT + .50*(ATAN3(DR,DZ,APT+BDMSLA)-BDMSLA) + PIQ2
      SNP    = SIN(AP)
      CSP    = COS(AP)
C     D      = SIN(AB-AP)
      D      = SNB*CSP-CSB*SNP
      IF(ABS(D).LT..01) GO TO 150
      XP     = XPT-ZBT(LB)
      YP     = YPT-RBT(LB)
      SS     = (YP*CSP-XP*SNP)/D
      F      = SS/SB
      IF(F.GE.1.0001) GO TO 140
      IF(F.GT.(-.0001) .OR. FGE1) GO TO 200
C      F .LE.-.0001
      GO TO 150
C      F .GE. 1.0001
  140 FGE1   = .TRUE.
  150 CONTINUE

C     FAILED TO FIND PROPER BOUNDARY INTERSECTION
      APTD   = APT*TODEG
      WRITE (6,1950) NAMBDY,XPT,YPT,APTD

C     FIRST OR LAST INTERVAL
      LB     = LB1
      F      = .1
      IF(.NOT.FGE1) GO TO 165
      LB     = LB2
      F      = .9
  165 DZ     = ZBT(LB+3)-ZBT(LB)
      DR     = RBT(LB+3)-RBT(LB)
      WRITE (6,1960)

  200 ANGCHD= ATAN3(DR,DZ,ANGBT(LB))
      F      = AMAX1(0.,AMIN1(F,1.))
      G      = 1.-F
      YPA    = ANGBT(LB)-ANGCHD
      YPB    = ANGBT(LB+3)-ANGCHD
```

```
      RZONLY= .FALSE.
      CALL BFI
      I      = (LB-LB10+3)/3
      FA     = F
      S1     = S1M
      XB     = ZBT(LB)+ZM
      YB     = RBT(LB)+RM
      RZONLY= .TRUE.
      RETURN

 1950 FORMAT(/1X61HERROR- THE INTERSECTION OF A L.E. OR T.E. ORTHOGONAL
     *WITH THE/6X14HBOUNDARY, BDY=A6,40H, WAS NOT FOUND,  THE L.E./T.E.
     *POINT IS/6X2HX=F10.5,3X2HY=F10.5,4X4HANG=F8.3,)
 1960 FORMAT(/6X58HTHE INTERSECTION POINT IS BEING PLACED IN AN END INTE
     *RVAL./6X24HEXECUTION WILL CONTINUE.)
      END
```

```
*DECK JOFCHN
      SUBROUTINE JOFCHN(CHN,JA,JB)
*JOFCHN         STREAMLINE INDEX FROM CHANNEL NAME              -JOFCHN-

C     INPUT-
C      CHN    = NAME OF CHANNEL
C      JA     = STREAMLINE FOR WHICH SEARCH WILL BE INITIATED

C     OUTPUT-
C      JA,JB = FIRST AND LAST INDEX OF STREAMLINES BELONGING TO CHN

       INTEGER CHN
      COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
     *                LO,LESTA,  LDUM(8),
     *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
     *                LEO,LEE, LRO,LRE,LRD
       DIMENSION      LIMITS(24)
       EQUIVALENCE    (LIMITS,LHO)
      COMMON /SLTAB / W(128),X2(128),SLCHN(128)
       INTEGER SLCHN

      LOGICAL         SECOND

      SECOND= .FALSE.
      J     = JA
   55 IF(CHN.NE.SLCHN(J)) GO TO 65
      IF(SECOND) GO TO 60
      SECOND= .TRUE.
      JA    = J
   60 JB    = J
      GO TO 70
   65 IF(SECOND) RETURN
   70 J     = J+1
      IF(J.LE.NJ) GO TO 55
      IF(.NOT.SECOND) CALL ERROR1
      RETURN
      END
```

```
*DECK OBI
      SUBROUTINE OBI(XPT,YPT,APT,NAMBDY,NAMCHN, I,FA,S1,XB,YB)
*OBI---          ORTHOGONAL-BOUNDARY INTERSECTION              -OBI-

C     INPUT-
C      XPT    = X-COOR OF PT ON THE ORTHOGONAL
C      YPT    = Y-COOR OF PT ON THE ORTHOGONAL
C      APT    = ANGLE OF SL PERPENDICULAR TO ORTHOGONAL
C      NAMBDY= BOUNDARY NAME
C      NAMCHN= NAME OF CHANNEL ADJACENT TO  NAMBDY

C     OUTPUT-
C      I      = INTERVAL OF ORTHOGONAL-BOUNDARY INTERSECTION
C      FA     = FRACTIONAL POSITION IN THE INTERVAL
C      S1     = ARC DISTANCE FROM BEGINNING OF THE INTERVAL
C      XB,YB  = COORDINATES OF THE INTERSECTION

C     BOUNDARY TABLE
C      INDEX- LB=LBDO,LBDE
C      LBNEXT= INCREMENT TO NEXT BOUNDARY
C      LBZ1   = INCREMENT TO THE FIRST BOUNDARY POINT (=0 BEFORE COALLATIO
C      CHNAME= CHANNEL WITH WHICH THE BOUNDARY DATA IS ASSOCIATED
C      UP     = T OR F FOR UPPER OR LOWER BOUNDARY
C      LEDEX  = RELATIVE INDEX OF L.E. POINT WHEN LOWER AND UPPER SURFACE
C               CONTOURS ARE CONNECTED
C      BDNAME,LBA,LBB=NAME AND INDEX LIMITS OF SPECIFIC BOUNDARY
C                     DATA WHEN BOUNDARIES ARE COALLATED
       COMMON /CHDATA/ BDT(1),LBNEXT(1),LBZ1(1),
      1                CHNAME(1),UP(1),LEDEX(1),
      2                ZBT(1),RBT(1),ANGBT(42)
         INTEGER       CHNAME
         LOGICAL       UP
         DIMENSION     BDNAME(1),LBA(1),LBB(1)
         EQUIVALENCE   (BDNAME,ZBT), (LBA,RBT), (LBB,ANGBT)
       COMMON /CBEAM2/ DR,DZ,YPA,YPB,F,G, DX,YQDX,ZM,RM,ANGM,CURVM,S1M,
      1                RZONLY, ANGCHD,SINTVL, YPASQ,YPAB,YPBSQ
         LOGICAL       RZONLY
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
         DIMENSION     LIMITS(24)
         EQUIVALENCE   (LIMITS,LHO)

       COMMON /CPI    / PI,TWOPI,PIQ2,PIQ4,TODEG,TORAD
       COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
         LOGICAL       ERR,ERRMAJ,INERR,PRERR

       LOGICAL         FGE1

C     DETERMINE INTERVAL INDEX LIMITS, LB1,LB2, OF -NAMBDY-
       LB     = LBF(NAMBDY)
       LB10   = LB+LBZ1(LB)
       LB20   = LB+LBNEXT(LB)-12
       LB1    = LB10
       LB2    = LB20
       IF(LEDEX(LB).NE.0) GO TO 105
C      BDMSLA= BOUNDARY MINUS STREAMLINE ANGLE
```

```
*DECK BWAKE
      SUBROUTINE BWAKE(JX,THK)
*BWAKE-         BUILD WAKE TABLE                                  -BWAKE-

C     INPUT-
C      JX     = WAKE STREAMLINE
C      THK    = T.E. THICKNESS

       COMMON /IXORIG/ LHO,LHE, LBDO,LBDL, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LOUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
        DIMENSION      LIMITS(24)
        EQUIVALENCE    (LIMITS,LHO)
       COMMON /SLTAB / W(128),X2(128),SLCHN(128)
        INTEGER SLCHN
C     TABLE OF WAKE DISPLACEMENT THICKNESS
C      INDEX- LW=LWO,LWE
       COMMON /CHDATA/        X2W(1),LWNEXT(1),S1W(47)
        DIMENSION      DST(1)
        EQUIVALENCE    (DST,S1W)
C      SUBTABLE ARRANGEMENT IS-
C       X2W,LWNEXT(=2+2N), S1W(1),S1W(2)...S1W(N), DST(1),DST(2),..DST(N)
C      X2W    = STREAMLINE COORDINATE
C      S1W    = DISTANCE ALONG STREAMLINE FROM T.E.
C      DST    = WAKE DISPLACEMENT THICKNESS AS A FUNCTION OF S1W

       IF(THK.EQ.0.) RETURN
       IF(LWE.GT.LWO) GO TO 110
       LW     = LWO
  110  X2W(LW)=X2(JX)
       S1W(LW)=0.
       S1W(LW+1)=10.*ABS(THK)
       S1W(LW+2)=S1W(LW+1)
       S1W(LW+3)=S1W(LW+2)+S1W(LW+2)
       DST(LW+4)=THK
       DST(LW+5)=0.
       DST(LW+6)=0.
       DST(LW+7)=0.
       N      = 4
       LWNEXT(LW)=2+N+N
       LW     = LW+LWNEXT(LW)
       LWE    = LW-1
       IF (THK.LT.0.) WRITE (6,1200) THK,X2(JX)
 1200  FORMAT(41H ***  ERROR - NEGATIVE T.E. THICKNESS OF ,F11.5,
      1        8H AT X12=,F7.3,1H.)
       RETURN
       END
```

```
*DECK FILL
      SUBROUTINE FILL(X,Y,NA,NB)
CFILL
C     LINEAR INTERPOLATION TO FIL VACANCIES IN INPUT LISTS
      COMMON /CBITS/BITS
      DIMENSION X(10),Y(10)
C     FIND IA,IB - VACANT REGION
      IA=NA+1
      IF(Y(IA-1).EQ.BITS) GO TO 99
    3 DO 4 I=IA,NB
      IF(Y(I).NE.BITS) GO TO 5
    4 CONTINUE
      IB=NB
      GO TO 7
    5 IB=I-1
      IF(I.EQ.IA) GO TO 12
C     FILL VACANCIES
      IF(Y(IB+1).NE.Y(IA-1)) GO TO 9
C     ALL VALUES THE SAME
    7 DO 8 II=IA,IB
    8 Y(II)=Y(IA-1)
      GO TO 12
C     INTERPOLATE
    9 DX = X(IB+1) - X(IA-1)
      DO 11 II=IA,IB
   11 Y(II) = (Y(IB+1)*(X(II)-X(IA-1)) + Y(IA-1)*(X(IB+1)-X(II)))/DX
C     GO BACK AND SEARCH FOR MORE REGIONS
   12 IA = IB+2
      IF(I.LT.NB) GO TO 3
   99 RETURN
      END
```

```
      LR2    = LRXF(LF)-LRDIF
      DO 1115 LR=LRO,LR2
      LRX    = LR+LRDIF
      IF(CHNA(LRX).EQ.BLANK .OR. JCHNA(LR).EQ.1) GO TO 1115
      NBLNK1= NBLNK1+1
      IF(J.EQ.0)  J=JCHNA(LR)
      IF(JCHNA(LR).NE.J) J=-1
 1115 CONTINUE
      NBLNK2= 0
      LR3    = LR2+1
      DO 1120 LR=LR3,LR4
      LRX    = LR+LRDIF
      IF(CHNA(LRX).EQ.BLANK .OR. JCHNA(LR).EQ.1) GO TO 1120
      NBLNK2= NBLNK2+1
      IF(JCHNA(LR).NE.J) J=-1
 1120 CONTINUE

C       SET STATIONS ABOVE T.E. TO NBLNK2
C       THE FLOW IS KNOWN IF NBLNK=J
      IF((NBLNK1+NBLNK2).EQ.0) GO TO 1140
      IF(NBLNK2.EQ.0) GO TO 1130
      JORDER(LF)=1
      DO 1125 LR=LR3,LR4
      LRX    = LR+LRDIF
 1125 IF(CHNA(LRX).NE.BLANK .AND. JCHNA(LR).NE.1) JCHNA(LR)=NBLNK2
C       SET STATIONS BELOW T.E. TO NBLNK1
 1130 IF(NBLNK1.EQ.0) GO TO 1136
      JORDER(LF)=2
      DO 1135 LR=LRO,LR2
      LRX    = LR+LRDIF
 1135 IF(CHNA(LRX).NE.BLANK .AND. JCHNA(LR).NE.1) JCHNA(LR)=NBLNK1

C      IS THE TOTAL FLOW KNOWN
 1136 IF(NBLNK1.EQ.0 .OR. NBLNK2.EQ.0) GO TO 1138
      JORDER(LF)=0
      IF(J.EQ.(-1)) GO TO 1140

C      INDEX TO NEXT T.E. ORTHOGONALS
 1138 NCHB(LF)=NBLNK1
      NCHA(LF)=NBLNK2
      X1BF(LF)=X1F(LF)
      X1AF(LF)=X1F(LF)
      LF     = LF+NFCOLS
      IF(LF.LE.LFE) GO TO 1112

C     ELIMINATE GAPS BETWEEN EQUIVALENCED TABLES
 1139 NMOVE  = LWE-LWO+1
      CALL MOVE(1, X2W(LWO),X2W(LTE+1),NMOVE,1)
      LWO    = LTE+1
      LWE    = LTE+NMOVE

      NMOVE  = LFE-LFO+1
      CALL MOVE(1, X1F(LFO),X1F(LWE+1),NMOVE,1)
      LFO    = LWE+1
      LFE    = LWE+NMOVE

      NMOVE  = LESTA-LO+1
      CALL MOVE(1, X1(LO),X1(LFE+1),NMOVE,1)
```

```
      LD     = LFE+1
      LESTA  = LFE+NMOVE

C     INITIALIZE  B
      CALL SETM(1,1./1024., B,NM)

      RETURN

C     ERROR-
 1140 WRITE (6,1141) LF
      CALL ERROR1

 1056 FORMAT(/2X60H*** ERROR- ONE CHANNEL MUST HAVE A FIXED FLOW RATE (V
     *ARY=F)./13X31HNO SUCH CHANNEL COULD BE FOUND./13X7HNCHTOT=I2,3X4HN
     *FA=I2,3X6HNCHBA=I2,)
 1560 FORMAT(1X47HERROR- CONNECTING EDGES WERE NOT FOUND FOR CHN=A6,22H
     *  (SUBROUTINE BLDTBS))
 1141 FORMAT(/1X61H***  PROBLEM ENCOUNTERED IN THE ORDERING OF FLOW ADJU
     *STMENTS./6X9HINDEX LF=I3,22H.  PLEASE SEEK ADVISE.,10H  (BLDTBS))
 1835 FORMAT(/1X47H***  ERROR- NEGATIVE RADIUS ENCOUNTERED, AXI=T.,
     *         10H  (BLDTBS))
      END
```

```
      X2CL(L)=X2CL(L+20)
      GO TO 950
  940 X2CL(L)=BITS
  950 L      = L+20
      GO TO 935

  960 L      = LO
      IF(X2CL(L).NE.BITS) GO TO 980
      M      = MLB(L)
      CALL GETIX
      X2CL(L)=X2(J)

C     BUILD WAKE DISPLACEMENT THICKNESS TABLE, /WAKETB/
  980 IF(LFE.LE.LFO) GO TO 1139
      LF     = LFO
  990 LBX    = LFB(LF)
      LAX    = LFA(LF)
      ANGTE  = S1F(LF)
C      THE MEAN T.E. ANGLE WAS TEMPORARILY STORED IN S1F
      M1     = MUB(LBX)
      M      = MLB(LAX)
      DZ21   = Z(M)-Z(M1)
      DR21   = R(M)-R(M1)
      THK    = DZ21*DZ21+DR21*DR21
      IF(THK.EQ.0.) GO TO 995
      DANG   = ATAN3(DR21,DZ21,ANGTE)-PIO2-ANGTE
      THK    = COS(DANG)*SQRT(THK)
  995 CALL GETIX
      CALL BWAKE(J,THK)
      LF     = LF+NFCOLS
      IF(LF.LT.LFE) GO TO 990

C     SEARCH FOR TRAILING EDGE ORTHOGONAL WHICH CONTROLS FLOW RATES
C      NFA    = NUMBER OF FLOW ADJUSTMENTS =NO. OF TE-S
      NFA    = (LFE+1-LFO)/NFCOLS
      NCHTOT= NJ/2
      FIXCHN = 0.
      LF     = LFO
 1005 IF(LF.GE.LFE) GO TO 1055
C      STATION BELOW T.E.
      L      = LFB(LF)
      JORDER(LF)=1

C      CHANNEL NAME (A SINGLE CHANNEL ABOVE OR BELOW T.E. IS REQD FOR  FI
C      PROGRAM MODIFICATION MAY 71, THE SINGLE CHN REQUIREMENT
C      FOR FIXCHN IS REMOVED.  NUMBER OF CHANNELS BELOW OR ABOVE
C      TE FOR FIXING THE FLOW MUST BE LESS OR EQUAL TO TOTAL
C      NUMBER OF CHANNELS MINUS NUMBER OF TE-S.
 1010 M      = MLB(L)
      CALL GETIX
      CHX    = SLCHN(J)
      JLB    = J
      M      = MUB(L)
      CALL GETIX
C      IF(CHX.NE.SLCHN(J)) GO TO 1040
      NCHBA = (J+1-JLB)/2
      IF(NCHBA.GT.(NCHTOT-NFA)) GO TO 1040
C      SEARCH CHDATA TABLE
```

198

```
 1015 LH     = LHO
 1020 IF(LH.GE.LHE) GO TO 1035
      IF(CHNAM(LH).EQ.CHX) GO TO 1030
      LH     = LH+LHNEXT(LH)
      GO TO 1020

 1030 FIXCHN = 1
      IF(VARY(LH)) GO TO 1040
 1035 FIXCHN= CHX
      WRITE (6,1031) FIXCHN
 1031 FORMAT(8H FIXCHN=,A6)
      GO TO 1060

C     STATION ABOVE T.E.
 1040 IF(JORDER(LF).EQ.2) GO TO 1050
      L      = LFA(LF)
      JORDER(LF)=2
      GO TO 1010
C     INDEX TO NEXT  X1F  IN FLOW ADJUSTMENT TABLE
 1050 LF     = LF+NFCOLS
      GO TO 1005

 1055 IF(FIXCHN.NE.1) GO TO 1060
      WRITE (6,1056) NCHTOT,NFA,NCHBA

C    ORDER THE FLOW ADJUSTMENT TABLE
C1060 IF(LF.EQ.LFO) GO TO 1100
 1060 LFF    = LF
      LFT    = LFO
 1070 IF(LFF.LE.LFT) GO TO 1100
      CALL MOVE(3, X1F(LFF),XX,NFCOLS,1,
     1             X1F(LFT),X1F(LFF),NFCOLS,1,
     2             XX,X1F(LFT),NFCOLS,1)
      LFF    = LFF-NFCOLS
      LFT    = LFT+NFCOLS
      GO TO 1070

C    DEFINE FLOW ADJUSTMENT ORDER, JORDER(LF)
C     FIND INDEX OF -FIXCHN- IN /ORTCHN/  AND  INITIALIZE
 1100 LR4    = LRO+LRD-3
      DO 1105 LR=LRO,LR4
      LR2    = LR+LRD
 1105 IF(CHNA(LR).EQ.FIXCHN .OR. CHNA(LR2).EQ.FIXCHN) GO TO 1110
 1110 CALL SETM(1,0, JCHNA,LRD-2)
      JCHNA(LR)=1

C     LOOP THROUGH FLOW ADJUSTMENT TABLE OF TE STATIONS, DETERMINE
C     IF FLOW IS KNOWN BELOW T.E. (JORDER=1), ABOVE T.E. (JORDER=2),
C     OR IF TOTAL FLOW ABOVE+BELOW T.E. IS KNOWN (JORDER=0).
C     JCHNA(LR)=1 INDICATES FLOW FOR THAT CHANNEL IS KNOWN, VALUES LARGE
C     THAN ONE INDICATE THAT THE TOTAL FLOW FOR THE SEVERAL CHANNELS
C     IS KNOWN.
      LF     = LFO
C      COUNT NUMBER OF CHNS BELOW T.E. AND ABOVE T.E. FOR WHICH FLOW RAT
C      IS NOT KNOWN, I.E.  JCHNA(LR).NE.1
 1112 J      = 0
      NBLNK1= 0
      LRDIF = LRF(LF)-LRO
```

```
      LRX2  = LRPRV+LRD-3
      DO 866 LRX=LRPRV,LRX2
  866 IF(CHNA(LRX).EQ.CHNX) GO TO 925
C      DID NOT FIND CHNX, SAVE LRPRV
      IF(LRPRSV.NE.0) CALL ERROR1
      LRPRSV= LRPRV

C     FIND UPSTREAM BOUNDARY WHICH INCLUDES CHANNEL CHNX
  870 LR    = LR+LRD
      CALL SETM(1,BLANK, LEDGE(LR),LRD)
      LRE   = LR+LRD-1
      LRPRV = LRD+LRD
      LRP1  = LRPRV
      LRP2  = LRP1+LRD-3
      DO 872 LRP=LRP1,LRP2
  872 IF(CHNA(LRP).EQ.CHNX) GO TO 873
      CALL ERROR1
  873 LR1   = LR+LRP-LRP1
      CHNA(LR1)=CHNX
      LR2   = LR1
      LRP1  = LRP
      LRP2  = LRP

C     SEARCH FOR CHANNELS BELOW CHNA(LR1)
  875 DO 876 LE1=LEO,LEE,10
  876 IF(NLE(LE1).NE.0 .AND. CHL(LE1).EQ.CHNA(LR1))GO TO 878
      GO TO 896
C      CHECK FOR BOTTOM CHANNEL
  878 IF(CHU(LE1).EQ.BLANK) GO TO 884
C      USE CHU(LE1) AS PART OF THE UPSTREAM BOUNDARY
  880 LRP1  = LRP1-1
      LR1   = LR1-1
      IF(CHU(LE1).EQ.CHNA(LRP1)) GO TO 882
      IF(LR1.GT.LR) GO TO 880
      GO TO 896
  882 CHNA(LR1)=CHU(LE1)
      GO TO 875

C     SEARCH FOR CHANNEL ABOVE LR2
  884 DO 888 LE2=LEO,LEE,10
  888 IF(NLE(LE2).NE.0 .AND. CHU(LE2).EQ.CHNA(LR2)) GO TO 892
      GO TO 896
C      CHECK FOR TOP CHANNEL
  892 IF(CHL(LE2).EQ.BLANK) GO TO 899
C      USE CHL(LE2) AS PART OF THE UPSTREAM BOUNDARY
  894 LRP2  = LRP2+1
      LR2   = LR2+1
      IF(CHL(LE2).EQ.CHNA(LRP2)) GO TO 898
      IF(LR2.LT.LRE) GO TO 894
  896 CALL ERROR1
C     REFER ALSO TO EFN 876, 882,888, FOR THE ERROR
  898 CHNA(LR2)=CHL(LE2)
      GO TO 884

  899 LE    = LE1
      UPT   = .FALSE.
      GO TO 672
```

```
C     TRAILING EDGE PT WITH ORTHOGONALS ON BOTH SIDES, BUILD DUMMY
C      LRPRV-LIST TO REPRESENT COALESCING OF UPPER AND LOWER STREAMS.
C       LOOK BACK FOR ORTHOG ON OTHER SIDE OF T.E.
  900 DO 904 LRP=LRO,LRE,LRD
  904 IF(LEDGE(LRP).EQ.LE) GO TO 908
  908 LEDGE(LRP)=-LEDGE(LRP)
      LRX1   = LRP
      LRX2   = LR
      LR     = LR+LRD
      LRX    = LR
      LRE    = LR+LRD-3
      CALL SETM(1,BLANK,LEDGE(LR),LRD)
      LEDGE(LR)=0
      LRPREV(LR)=LRX2
  910 IF(CHNA(LRX1).NE.BLANK) CHNA(LRX)=CHNA(LRX1)
      IF(CHNA(LRX2).NE.BLANK) CHNA(LRX)=CHNA(LRX2)
      LRX1   = LRX1+1
      LRX2   = LRX2+1
      LRX    = LRX+1
      IF(LRX.LE.LRE) GO TO 910
      LRE    = LRE+2

C     BUILD FLOW ADJUSTMENT TABLE, /CADJWF/
      LM1    = L-20
      X1F(LF)=X1(LM1)
      X2F(LF)=X2CL(LM1)
      S1F(LF)=ANGE(LE)
      DO 911 LM2=LO,LESTA,20
  911 IF(X1(LM2).EQ.X1(LM1)) GO TO 912
  912 IF(UPT) GO TO 913
      LFB(LF)=LM2
      LFA(LF)=LM1
      LRXF(LF)=LR1-1+LRD
      GO TO 914
  913 LFB(LF)=LM1
      LFA(LF)=LM2
      LRXF(LF)=LR2+LRD
  914 LRF(LF)=LR
      VNR(LF)= 0.
      LF     = LF+NFCOLS
      LFE    = LF-1
      GO TO 920

C     DOWNSTREAM BOUNDARY, ARE ALL L.E. ORTHOGONAL COMPLETED
  915 IF(LRPRV.NE.0) GO TO 925
      IF(LRPRSV.EQ.0) GO TO 930
      LRPRV = LRPRSV
      GO TO 925

  920 LRPRV = LR
  925 LR     = LR+LRD
      GO TO 640

C*** RELOCATE CONTROL STREAMLINE, X2CL, TO THE FIRST PRIMARY OF REGION
  930 L      = LO
  935 IF((L+20).GE.LESTA) GO TO 960
      IF(X1(L+20).LE.X1(L)) GO TO 940
      IF(X2CL(L+20).EQ.BITS) GO TO 950
```

206

```
      MD     = 0
      ISTAG = 0
      CALL SAVIX

C     ADD CHANNEL FLOWS FOR LATER INTERPOLATION OF SL POSITION
C     IF NOT AN UPSTREAM BOUNDARY, USE UPSTREAM AREAS IN PLACE OF FLOW.
C     -USE  CURV  FOR STORAGE
      WSL(M)= 0.
      IF(M.EQ.M1) GO TO 830
      WSL(M)= WSL(M-1)+W(J)
      IF(MSAV.EQ.0) GO TO 830
      AREA  = SQRT((R(MU)-R(MUM1))*(R(MU)-R(MUM1)) +
     1             (Z(MU)-Z(MUM1))*(Z(MU)-Z(MUM1)))
      IF(AXIA) AREA=(R(MU)+R(MUM1))*AREA
      WSL(M)= WSL(M-1)+AREA

  830 MM     = MM+1
      MUM1   = MU
      IF(JSL.EQ.JNXT) GO TO 835
      JSL    = JNXT
      GO TO 810

C     INCREMENT TO NEXT CHANNEL
  835 LRN    = LRN+1
      IF(LRN.LE.LR2) GO TO 805

C    INTERPOLATE FOR COORDINATES
         IF(.NOT.AXIA .OR. R(M1).GE.0.) GO TO 836
         WRITE (6,1835)
         CALL ERROR1
  836 DZ21   = Z(M2)-Z(M1)
      DR21   = R(M2)-R(M1)
      DRSQ21= DR21*(R(M2)+R(M1))
      RM1SQ  = R(M1)*R(M1)
      S2(M1)= 0.
      S2(M2)= SQRT(DZ21*DZ21+DR21*DR21)
      VM(M1)= VMINIT
      VM(M2)= VMINIT
      MP     = M1+1
      MM     = M2-1
      IF(MM.LT.MP) GO TO 840
      DO 838 M=MP,MM
      VM(M)  = VMINIT
      F      = (WSL(M)-WSL(M1))/(WSL(M2)-WSL(M1))
      Z(M)   = Z(M1)+F*DZ21
      R(M)   = R(M1)+F*DR21
      S2(M)  = F*S2(M2)
      IF(.NOT.AXIA) GO TO 838
      R(M)   = SQRT(RM1SQ+F*DRSQ21)
      S2(M)  = SQRT((R(M)-R(M1))*(R(M)-R(M1))+(F*DZ21)*(F*DZ21))
  838 CONTINUE

C    FINISH OUT STATION TABLE
C    CHECK FOR L.E., T.E. OR SHARP CORNER
C     LE     = INDEX OF PT IN LETEPT-TABLE
C     NLETE  = 0 IS A SHARP CORNER
  840 X1(L)  = 8.*FLOAT((LE+1-LE0)/10)
      LNEXT(L)=20
```

201

```
      TYPELB(L)=SOLID
      TYPEUB(L)=SOLID
      X2CL(L)=BITS
      IF(NLETE.EQ.1) GO TO 848
      IF(UPT) GO TO 842
C      UPT=F
      X2CL(L)=X2(J1)
      M      = MLB(L)
      GO TO 843
C      UPT=T
  842 X2CL(L)=X2(J2)
      M      = MUB(L)
  843 CALL GETIX
      IF(NLE(LE).NE.2) GO TO 845
      ISTAG = 1
      IF(UPT) GO TO 844
      TYPELB(L)=HLE
      GO TO 845
  844 TYPEUB(L)=HLE
  845 IF(NTE(LE).NE.2) GO TO 847
      ISTAG = 2
      IF(UPT) GO TO 846
      TYPELB(L)=HTE
      GO TO 847
  846 TYPEUB(L)=HTE
  847 IF(NLETE.EQ.0) ISTAG=MCL(L)
      CALL SAVIX
      IF(ISTAG.EQ.1) VM(M)=0.
  848 VMB(L)= VMINIT
      DWDV(L)=0.
      VCL(L)= VMB(L)
      PRIM(L)=.TRUE.
      M      = MUB(L)+1
      L      = L+LNEXT(L)
      LESTA = L-1

C*    INDEX TO NEXT ORTHOGONAL
C     LOOK FOR ORTHOGONALS TO BE PLACED ABOVE L.E. POINTS
C      IF THIS IS A DOWNSTREAM BOUNDARY OR LOWER T.E. ORTHOG
  850 IF(NTE(LE).EQ.0) GO TO 920
      IF(NTE(LE).EQ.1) GO TO 855
C      NTE(LE)=2
      IF(NUSED(LE).EQ.2) GO TO 900
  855 LRX    = LR
  860 LRX    = LRPREV(LRX)
C      LRPREV= BLANK FOR UPSTREAM OR DUMMY ORTCHN-LISTS
      IF(LRPREV(LRX).EQ.IBLANK) GO TO 862
      IF(LEDGE(LRX).LE.0) GO TO 860
      LRPRV = LRPREV(LRX)
      GO TO 864
  862 LRPRV = LRPRSV
      LRPRSV= 0

C     IS THE CHANNEL ON THE OTHER SIDE OF THE T.E. IN THE LRPRV-LIST
  864 IF(NTE(LE).NE.2) GO TO 915
      CHNX   = CHU(LE)
      IF(UPT) CHNX=CHL(LE)
      IF(LRPRV.EQ.0) GO TO 870
```

```
      S1LB(L)=SINTVL
      IUB(L)= 1
      FUB(L)= 0.
      S1UB(L)=0.
  715 Z(M1) = XE(LE1)
      R(M1) = YE(LE1)
      Z(M2) = XE(LE2)
      R(M2) = YE(LE2)
      GO TO 800

C     FIND LE OR TE ORTHOGONAL LOWER BOUNDARY INTERSECTION
C      PLACE DATA IN STATION TABLE
C       USE LETEPT-TABLE TO DETERMINE NAME OF UPPER BOUNDARY
  720 IF(NLETE.EQ.2 .OR. NLETE.EQ.0) GO TO 722
      CALL ERROR1
  722 IF(.NOT.UPT) GO TO 740
      DO 725 LE1=LEO,LEE,10
  725 IF(CHL(LE1).EQ.CHNA(LR1)) GO TO 726
  726 NAMELB(L)=BDL(LE1)
      NAMEUB(L)=BDU(LE)
      CALL OBI(XE(LE),YE(LE),ANGE(LE),BDL(LE1),CHL(LE1),
     1          ILB(L),FLB(L),S1LB(L), Z(M1),R(M1))
C      SEEK POINTER TO BOUNDARY TABLE
      LB    = LBF(NAMEUB(L))
      IRET  = 1
      IF(NTE(LE).NE.2) GO TO 728
C       TRAILING EDGE
      IV    = 1
      LB    = LB+LBZ1(LB)
      GO TO 733
C       LEADING EDGE OR CORNER
  728 LB1   = LB+LBZ1(LB)
      LB2   = LB+LBNEXT(LB)-9
      IV    = 1
      DO 730 LB=LB1,LB2,3
      IF(ZBT(LB).EQ.XE(LE) .AND. RBT(LB).EQ.YE(LE)) GO TO 732
  730 IV    = IV+1
      CALL ERROR1
C       TEMPORARILY STORE SHARP CORNER INDICATION IN MCL(L) (I.E. ANGLE
C       JUMP OF MORE THAN 0.5 DEG.)
  732 MCL(L)= 2
      IF(NLETE.EQ.0 .AND. ABS(ANGBT(LB)-ANGBT(LB+3)).GT..0087) MCL(L)=1
      IF(IRET) 733,753,733
  733 IUB(L)= IV
      FUB(L)= 0.
      S1UB(L)=0.
      Z(M2) = ZBT(LB)
      R(M2) = RBT(LB)
      GO TO 800

C     FIND LE OR TE ORTHOGONAL UPPER BOUNDARY INTERSECTION
C      PLACE DATA IN STATION TABLE
  740 DO 745 LE2=LEO,LEE,10
  745 IF(CHU(LE2).EQ.CHNA(LR2)) GO TO 747
  747 NAMELB(L)=BDL(LE)
      NAMEUB(L)=BDU(LE2)
      CALL OBI(XE(LE),YE(LE),ANGE(LE),BDU(LE2),CHU(LE2),
     1          IUB(L),FUB(L),S1UB(L), Z(M2),R(M2))
```

```
C       SEEK POINTER TO BOUNDARY TABLE
        LB      = LBF(NAMELB(L))
        IRET    = 0
        IF(NTE(LE).NE.2) GO TO 728
C        TRAILING EDGE
        LB2     = LB+LBNEXT(LB)-9
        ILB(L)= (LB2-(LB+LBZ1(LB)))/3
        FLB(L)= 1.
        CALL BARC(LB2-3)
        SILB(L)=SINTVL
        LB      = LB2
        GO TO 757
C        LEADING EDGE OR CORNER
  753 ILB(L)= IV
        FLB(L)= 0.
        SILB(L)=0.
  757 Z(M1) = ZBT(LB)
        R(M1) = RBT(LB)

C*    ADD NEW FIELD POINTS ALONG EXISTING STREAMLINES
C     GIVEN-
C       STA-TAB INDEX L AND LIMITS ON FIELD INDEX MLB,MUB
C       COORDINATES OF FIRST AND LAST NEW PTS IN FIELD TABLE
C       MARKED CHANNELS IN ORTCHN TABLE BETWEEN LR1,LR2
C       STREAMLINE TABLE
  800 MSAV    = MO
C       MSAV    = 0 INDICATES UPSTREAM BOUNDARY
        IF(NLE(LE).EQ.1) MSAV=0
        J1      = 1
        CALL JOFCHN(CHNA(LR1),J1,JX)
        CALL JOFCHN(CHNA(LR2),JX,J2)
C       J1,J2 ARE SL INDEX LIMITS

C       BEGIN LOOP THROUGH CHANNELS,  2 SLS PER CHANNEL
        LRN     = LR1
        MM      = M1
        JSL     = J1
  805 IF(CHNA(LRN).EQ.BLANK) GO TO 835
        CALL JOFCHN(CHNA(LRN),JSL,JNXT)

C       FIND UPSTREAM FIELD PT, PUT IN DOWNSTREAM POINTER
  810 J       = JSL
        IF(MSAV) 812,828,812
  812 IV      = 0
  815 DO 820 M=MSAV,NM
        CALL GETIX
  820 IF(J.EQ.JSL .AND. MD.EQ.0) GO TO 825
        IF(IV.NE.0) CALL ERROR1
        MSAV    = MO
        IV      = 1
        GO TO 815
  825 MSAV    = M
        MD      = MM
        CALL SAVIX

C       SAVE DATA FOR CURRENT FIELD PT
  828 M       = MM
        MU      = MSAV
```

204

```
C       EACH ORTHOGONAL.
  639 LRPRV = LRO
      LRPRSV= 0
      M     = MO
      L     = LO
      LF    = LFO
      TTESQ = TTE*TTE

C*   CONSIDER MARKED CHANNELS ON LINE LR=LRPRV IN /ORTCHN/
C     FIND INDEX OF FIRST AND LAST ACTIVE (NON-BLANK) CHANNEL
  640 LRP1  = LRPRV
      LRP2  = LRPRV+LRD-3
  642 IF(CHNA(LRP1).NE.BLANK) GO TO 644
      LRP1  = LRP1+1
      IF(LRP1.LE.LRP2) GO TO 642
      CALL ERROR1
  644 IF(CHNA(LRP2).NE.BLANK) GO TO 646
      LRP2  = LRP2-1
      GO TO 644

C     FIND INDEX-LE OF NEXT LE-TE PT IN LRPRV-CHANNELS
  646 LE    = LFO
  648 IF(NUSED(LE)-NLE(LE)-NTE(LE)) 650,654,654
  650 LEONCE= NUSED(LE)
      IF(NTE(LE).NE.0) LEONCE=0
      LRP   = LRP1
  652 IF(CHNA(LRP).EQ.BLANK) GO TO 653
      IF(CHNA(LRP).EQ.CHU(LE) .AND. LEONCE.LE.0) GO TO 660
      IF(CHNA(LRP).EQ.CHL(LE)) GO TO 665
  653 LRP   = LRP+1
      IF(LRP.LE.LRP2) GO TO 652
  654 LE    = LE+10
      IF(LE.LE.LEE) GO TO 648
C      NO MORE POINTS
       CALL ERROR1

C      LE IS UPPER BOUNDARY POINT (LOWER ORTHOGONAL)
  660 LRP2  = LRP
      UPT   = .TRUE.
      GO TO 670

C      LE IS LOWER BOUNDARY POINT (UPPER ORTHOGONAL)
  665 LRP1  = LRP
      UPT   = .FALSE.

C    MARK CHANNEL NAMES OF THE NEW ORTHOGONAL ON LINE LR
  670 CALL SETM(1,BLANK, LEDGE(LR),LRD)
      LR1   = LR + LRP1-LRPRV
      LR2   = LR + LRP2-LRPRV
      CALL MOVE(1, CHNA(LRP1),CHNA(LR1),LR2-LR1+1,1)
      LRE   = LR+LRD-1

C    UPDATE USED LETEPT COUNT AND SET POINTERS FOR LINE-LR
  672 NUSED(LE)=NUSED(LE)+1
      LRPREV(LR)=LRPRV
      LEDGE(LR)=LE
      NLETE = NLE(LE)+NTE(LE)
      IF(NLETE-NUSED(LE).EQ.0) LEDGE(LR)=-LEDGE(LR)
```

```
C     COUNT NUMBER OF CHANNELS, SET FIELD TABLE LIMITS
      NCHNA = 0
      DO 675 LRX=LR1,LR2
  675 IF(CHNA(LRX).NE.BLANK) NCHNA=NCHNA+1
      M1    = M
      MLB(L)= M1
      M2    = M1+NCHNA+NCHNA-1
      MUB(L)= M2

      NM    = M2
      LESTA = L+20

C     IF UPSTREAM OR DOWNSTREAM BOUNDARY, SEARCH FOR OTHER EDGE
      IF(NLE(LE).EQ.1) GO TO 679
      IF(NTE(LE).EQ.1) GO TO 681
      GO TO 720
  679 LX    = 0
      GO TO 682
  681 LX    = 1
  682 IF(.NOT.UPT) GO TO 690

C     FIND LOWER EDGE PT
  684 DO 686 LE1=LEO,LEE,10
      LEX   = LE1+LX
  686 IF(NLE(LEX).EQ.1 .AND. CHL(LE1).EQ.CHNA(LR1)) GO TO 688
      CALL ERROR1
  688 LE2   = LE
      NUSED(LE1)=NUSED(LE1)+1
      GO TO 700

C     FIND UPPER EDGE PT
  690 DO 692 LE2=LEO,LEE,10
      LEX   = LE2+LX
  692 IF(NLE(LEX).EQ.1 .AND. CHU(LE2).EQ.CHNA(LR2)) GO TO 694
      CALL ERROR1
  694 LE1   = LE
      NUSED(LE2)=NUSED(LE2)+1

C*    PLACE UPSTREAM OR DOWNSTREAM BOUNDARY DATA INTO STATION TABLE
  700 NAMELB(L)=BDL(LE1)
      NAMEUB(L)=BDU(LE2)
      IF(NTE(LE).EQ.1) GO TO 710
C      UPSTREAM BOUNDARY
      ILB(L)= 1
      FLB(L)= 0.
      S1LB(L)=0.
      LB    = LBF(NAMEUB(L))
      IUB(L)= (LBNEXT(LB)-9-LBZ1(LB))/3
      FUB(L)= 1.
      CALL BARC(LB+LBNEXT(LB)-12)
      S1UB(L)=SINTVL
      GO TO 715
C      DOWNSTREAM BOUNDARY
  710 LB    = LBF(NAMELB(L))
      ILB(L)= (LBNEXT(LB)-9-LBZ1(LB))/3
      FLB(L)= 1.
      CALL BARC(LB+LBNEXT(LB)-12)
```

```
C     LINE UP THE L.E. AND T.E. CONNECTED CHANNELS IN THE SAME COLUMN
C       LRL    = INDEX OF CHANNEL IN FIRST LINE (L.E. CONNECTED CHANNELS)
C       LRT    = INDEX OF CHANNEL IN SECOND LINE (T.E. CONNECTED CHANNELS)
        LRL    = LR1
        LRT    = LR3
        GO TO 588
  570 IF(LRE.LT.LRT) GO TO 578
        DO 575 LRX=LRT,LRE
  575 IF(CHNA(LRX).EQ.CHNA(LRL)) GO TO 580

C       LRL-CHANNEL NOT INCLUDED IN SECOND LINE, PUT IN BLANK SPACE.
        CALL MOVE(1, CHNA(LRT),CHNA(LRT+1),LRT-LRE-1,1)
  578 LRE    = LRE+1
        CHNA(LRT)=BLANK
        GO TO 586

C       LRX MATCHES LRL, PUT IN LRX-LRT BLANKS BEFORE LRL
  580 LDR    = LRX-LRT
        IF(LDR) 582,586,582
  582 LRTO   = LRL+LDR
        CALL MOVE(1, CHNA(LRL),CHNA(LRTO),LRL-LRE-1,1)
        LRE    = LRE+LDR
        LRT    = LRT+LDR
        LR2    = LR2+LDR
  584 CHNA(LRL)=BLANK
        LRL    = LRL+1
        LRT    = LRT+1
        IF(LRL.LT.LRTO) GO TO 584
C         IF NO CHANNELS ON FIRST LINE,  SET FIRST VALUE TO THAT OF SECOND
        IF(LR2-LDR.LT.LR1) CHNA(LR1)=CHNA(LR2+1)

  586 LRL    = LRL+1
        LRT    = LRT+1
  588 IF(LRL.LE.LR2) GO TO 570
        IF(LRT.GT.LRE) GO TO 600
        LDR    = LRE-LRT+1
        GO TO 582

C     DEFINE ORTCHN-TABLE INCREMENT, LRD
  600 LRD    = LR2-LRO+3
        CALL MOVE(1, CHNA(LR2+1),CHNA(LR2+3),LR2-LRE,1)
        LRE    = LRE+4
        LEDGE(LRO)=IBLANK
        LRPREV(LRO)=IBLANK
        LRPRV = LRO
        LR     = LRO+LRD
        LEDGE(LR)=IBLANK
        LRPREV(LR)=IBLANK
        LR     = LR+LRD
        IF(ERR) CALL ERROR1

C*    BUILD STREAMLINE TABLE
        NJ     = 0
        LRL    = LRO
        LRT    = LRO+LRD
        X2SAV1= 0.
        X2SAV2= 0.
```

```
      DAREA = 0.
C       SEARCH FOR FIRST COMMON CHANNEL
  602 LRX1  = LRL
      LRX2  = LRL+LRD
      NBLNK1= 0
      NBLNK2= 0
  605 IF(CHNA(LRX1).EQ.CHNA(LRX2)) GO TO 610
      IF(CHNA(LRX1).NE.BLANK) NBLNK1=NBLNK1+1
      IF(CHNA(LRX2).NE.BLANK) NBLNK2=NBLNK2+1
      LRX1  = LRX1+1
      LRX2  = LRX2+1
      IF(LRX1.LE.LR2) GO TO 605

  610 DX2   = 8.*AMAX0(NBLNK1,NBLNK2)
      IF(DX2.EQ.0.) GO TO 620
      IF(NBLNK1.NE.0) DEL1=DX2/FLOAT(NBLNK1)
      IF(NBLNK2.NE.0) DEL2=DX2/FLOAT(NBLNK2)
  612 IF(CHNA(LRL).EQ.BLANK) GO TO 615
      CHX   = CHNA(LRL)
      X2(NJ+1)=X2SAV1
      X2SAV1= X2SAV1+DEL1
      X2(NJ+2)=X2SAV1
      GO TO 625
  615 CHX   = CHNA(LRT)
      X2(NJ+1)=X2SAV2
      X2SAV2= X2SAV2+DEL2
      X2(NJ+2)=X2SAV2
      GO TO 625
  620 CHX   = CHNA(LRL)
      X2(NJ+1)=X2(NJ)
      IF(NJ.EQ.0) X2(NJ+1)=0.
      X2(NJ+2)=X2(NJ+1)+8.
      X2SAV1=X2(NJ+2)
      X2SAV2=X2(NJ+2)
  625 SLCHN(NJ+1)=CHX
      SLCHN(NJ+2)=CHX
      W(NJ+1)=0.
          DO 630 LE1=LE0,LEE,10
  630     IF(NLE(LE1).NE.0 .AND. CHL(LE1).EQ.CHX) GO TO 632
  632     DO 635 LE2=LE0,LEE,10
  635     IF(NLE(LE2).NE.0 .AND. CHU(LE2).EQ.CHX) GO TO 637
  637     AREA  = YE(LE2)-YE(LE1)
          IF(AXIA) AREA=AREA*PI*(YE(LE2)+YE(LE1))
C         FOR INLET CONF. SAVE HIGHLIGHT AREA SO EXTERNAL AREA
C         MAY BE CORRECTED BY DIFF BET HIGHLIGHT AND CAPTURE AREAS.
          AREASV= AREA
          IF(CHNA(LRL).NE.BLANK) AREA=AREA+DAREA
          CALL BCONV(CHX,LT,AREA)
          IF(CHNA(LRL).NE.BLANK) DAREA=DAREA+AREASV-AREA
          LT    = LT+LPSI(LT)+NPT(LT)-1
      W(NJ+2)=XCH(LT)
      NJ    = NJ+2
      LRL   = LRL+1
      LRT   = LRT+1
      IF(LRL.GT.LR2) GO TO 639
      IF(LRL-LRX1) 612,620,602

C**  BEGIN LOOP FOR BUILDING CHANNEL LIST, STATION TABLE AND FIELD DATA
```

```
      EQUIVALENCE   (GAM,FGT,NAMEUB), (NR,FGP,IUB), (NC,FGR,FUB)
      EQUIVALENCE   (TAB(1),AREATB,S1UB), (TAB(2),VMB), (TAB(3),DWDV)
      EQUIVALENCE   (TAB(4),X2CL), (TAB(5),VCL), (TAB(6),MCL)

      INTEGER        CHNX,CHX,FIXCHN,HLE,HTE,SOLID
      LOGICAL        UPT

      DATA SOLID/5HSOLID/, HLE/2HLE/, HTE/2HTE/

C     USE INPUT SPACERS TO SET TABLE ORIGINS
      LTE   = LBDE
      LTO   = LTE+1
      LWE   = LTE+MAXLT
      LWO   = LWE+1
      LFE   = LWE+MAXLW
      LFO   = LFE+1
      LESTA = LFE+MAXLF
      LO    = LESTA+1

C     ASSUMED INITIAL FIELD VELOCITY
      VMINIT= .5 * SQRT(GAMA*RGA*TTA)

C**   BUILD ORTHOGONAL-CHANNEL TABLE

C*    BUILD ORDERED LIST OF CHANNELS FROM L.E. CONNECTIONS
C      SEARCH FOR THE FIRST LEADING EDGE PT (NLE=2 IN LETEPT-TABLE)
      LR3   = LRO
      LRE   = LR3-1
      LX    = 0
      DO 505 LE=LEO,LEE,10
  505 IF(NLE(LE).EQ.2) GO TO 510
C      NO L.E. PTS
      GO TO 535

C     LE=FIRST EDGE PT, FIND CONNECTING CHANNELS
  510 CHNA(LR3)=CHU(LE)
      CHNA(LR3+1)=CHL(LE)
      LRE   = LR3+1

C     SEARCH FOR CHANNELS BELOW CHNA(LR3)
  515 DO 517 LE3=LEO,LEE,10
      LEX   = LE3+LX
  517 IF(NLE(LEX).NE.0 .AND. CHL(LE3).EQ.CHNA(LR3)) GO TO 520
      WRITE (6,1560) CHNA(LR3)
      CALL ERROR1

C     CHECK FOR BOTTOM CHANNEL
  520 IF(CHU(LE3).EQ.BLANK) GO TO 525

C     MOVE CHU(LE3) BELOW CHNA(LR3)
      CALL MOVE(2, CHNA(LR3),CHNA(LR3+1),LR3-LRE-1,1,
     1             CHU(LE3),CHNA(LR3),1,1)
      LRE   = LRE+1
      GO TO 515

C     SEARCH FOR CHANNELS ABOVE CHNA(LRE)
  525 DO 530 LE4=LEO,LEE,10
      LEX   = LE4+LX
```

```
  530 IF(NLE(LEX).NE.0 .AND. CHU(LE4).EQ.CHNA(LRE)) GO TO 532
      WRITE (6,1560) CHNA(LRE)
      CALL ERROR1

C     CHECK FOR TOP CHANNEL
  532 IF(CHL(LE4).EQ.BLANK) GO TO 535

C     MOVE CHL(LE4) ABOVE CHNA(LRE)
      LRE    = LRE+1
      CHNA(LRE)=CHL(LE4)
      GO TO 525

C    REPEAT THE ABOVE FOR THE TRAILING EDGE
  535 IF(LX.EQ.1) GO TO 545
      LR1    = LR3
      LR2    = LRE
      LE1    = LE3
      LE2    = LE4
      LR3    = LR2+1
      LX     = 1
C     LX     = 1  TO PICK UP NTE(LE3) RATHER THAN NLE(LE3)

C     SEARCH FOR THE LAST T.E. PT
      LE     = LEE-9
  540 IF(NTE(LE).EQ.2) GO TO 510
      LE     = LE-10
      IF(LE.GE.LEO) GO TO 540

C    NO L.E. OR T.E. PTS
  545 IF(LRE-LR1) 547,555,555
  547 LE     = LEO
      IF(CHL(LE).NE.BLANK) GO TO 550
      IF(CHU(LE).NE.BLANK) GO TO 552
      CALL ERROR1
  550 CHNA(LR1)=CHL(LE)
      GO TO 554
  552 CHNA(LR1)=CHU(LE)
  554 LR2    = LR1
      LR3    = LR2+1
      CHNA(LR3)=CHNA(LR1)
      LRE    = LR3

C    CHECK FOR EXTRA CHANNELS IN LETEP-TABLE
  555 LE     = LEO
  556 IF(CHL(LE).EQ.BLANK) GO TO 558
      CHX    = CHL(LE)
      LX     = 0
      GO TO 560
  558 IF(CHU(LE).EQ.BLANK) GO TO 564
      CHX    = CHU(LE)
      LX     = 1
  560 DO 561 LR=LR1,LRE
  561 IF(CHNA(LR).EQ.CHX) GO TO 562
      ERR    = .TRUE.
      WRITE (6,1560) CHX
  562 IF(LX) 564,558,564
  564 LE     = LE+10
      IF(LE.LT.LEE) GO TO 556
```

```
C      LRD    = NUMBER OF CHANNELS PLUS ONE,   LR INDEX INCREMENT
C      LEDGE  = INDEX OF THE ORTHOGONAL POINT IN THE LETEPT-TABLE
C      LRPREV= POINTER OF LINE OF UPSTREAM CHANNELS IN ORTCHN-TABLE
C      CHNA   = CHANNEL NAMES
       COMMON /ORTCHN/ LEDGE(1),LRPREV(1),CHNA(479)
        INTEGER CHNA
        DIMENSION       JCHNA(1)
        EQUIVALENCE     (JCHNA,CHNA)
C     STREAMLINE TABLE
       COMMON /SLTAB / W(128),X2(128),SLCHN(128)
        INTEGER SLCHN
       COMMON /CBITS / BITS,BLANK
        EQUIVALENCE     (IBLANK,BLANK)
        INTEGER         BLANK
       COMMON /CPI   / PI,TWOPI,PIQ2,PIQ4,TODEG,TORAD
       COMMON /CTABPR/ ITAB
       COMMON /ERASE / XX(1),YY,ANGG,NL,NT,CNL,CNU,BNL,BNU,NZERO
       COMMON /SPACER/ MAXLH,MAXLT,MAXLF,MAXLW
       COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
        LOGICAL         ERR,ERRMAJ,INERR,PRERR

C     CHANNEL INPUT DATA TABLE
C      INDEX- LH=LHO,LHL
       COMMON /CHDATA/ CHNAM(1),LHNEXT(1),WTFLOW(1),TTO(1),PTO(1),
      1                TSO(1),PSO(1),MACHO(1),AO(1),VARY(1),
      2                RG(1),GAM(1), NR(1),NC(1),TAB(6),
      4                BB(75)
        LOGICAL         VARY
        INTEGER CHNAM
        DIMENSION       VO(1)
        REAL            MACHO
        EQUIVALENCE     (VO,MACHO)
C     BOUNDARY TABLE
C      INDEX- LB=LBDO,LBDE
C      LBNEXT= INCREMENT TO NEXT BOUNDARY
C      LBZ1   = INCREMENT TO THE FIRST BOUNDARY POINT (=0 BEFORE COALLATIO
C      CHNAME= CHANNEL WITH WHICH THE BOUNDARY DATA IS ASSOCIATED
C      UP     = T OR F FOR UPPER OR LOWER BOUNDARY
C      LEDEX  = RELATIVE INDEX OF L.E. POINT WHEN LOWER AND UPPER SURFACE
C               CONTOURS ARE CONNECTED
C      BDNAME,LBA,LBB=NAME AND INDEX LIMITS OF SPECIFIC BOUNDARY
C                     DATA WHEN BOUNDARIES ARE COALLATED
       DIMENSION        BDT(1),LBNEXT(1),LBZ1(1),
      1                 CHNAME(1),UP(1),LEDEX(1),
      2                 ZBT(1),RBT(1),ANGBT(42)
        LOGICAL         UP
        INTEGER BDT,CHNAME,BDNAME
        DIMENSION       BDNAME(1),LBA(1),LBB(1)
        EQUIVALENCE     (BDNAME,ZBT), (LBA,RBT), (LBB,ANGBT)
C     TABLE OF CONVECTED PROPERTIES
C      INDEX- LT=LTO,LTE
C      CH     = CHANNELNAME
C      LTNEXT= INDEX INCREMENT TO THE NEXT CHANNEL
C      LPSI   = RELATIVE LOCATION OF PSI LIST
C      NPT    = NO. OF PSI, TT, PT AND RCU VALUES
C      LTT    = RELATIVE LOCATION OF TT LIST
C      LPT    = RELATIVE LOCATION OF PT LIST
C      LRCU   = RELATIVE LOCATION OF RCU LIST
```

```
      DIMENSION       CH(1),LTNEXT(1),NPT(1),LPSI(1),LTT(1),LPT(1),
     1                LRCU(1),
     2                CRG(1),CPGJ(1),C2CP(1),QGAM(1),FGT(1),FGP(1),
     3                FGR(1),AREATB(485)
        DIMENSION     XCH(1)
        EQUIVALENCE   (CH,XCH)
C     TABLE OF WAKE DISPLACEMENT THICKNESS
C      INDEX- LW=LWO,LWE
      DIMENSION                X2W(1),LWNEXT(1),S1W(47)
        DIMENSION     DST(1)
        EQUIVALENCE   (DST,S1W)
C      SUBTABLE ARRANGEMENT IS-
C       X2W,LWNEXT(=2+2N), S1W(1),S1W(2)...S1W(N), DST(1),DST(2),..DST(N)
C      X2W    = STREAMLINE COORDINATE
C      S1W    = DISTANCE ALONG STREAMLINE FROM T.E.
C      DST    = WAKE DISPLACEMENT THICKNESS AS A FUNCTION OF S1W
C     FLOW ADJUSTMENT TABLE
C      INDEX- LF=LFO,LFE
C      NFCOLS= 8
C      X1F    = ORTHOGONAL COORDINATE
C      X2F    = STREAMLINE COORDINATE OF SL EMINATING FROM T.E.
C      X1BF   = X1-COORDINATE OF CHOKE STATION OF FLOW BELOW T.E.
C      X1AF   = X1-COORDINATE OF CHOKE STATION OF FLOW ABOVE T.E.
C      S1F    = S1-COORDINATE OF T.E. (UPPER SURFACE). THIS ITEM
C               IS USED WHEN INTERPOLATING FOR WAKE DELTA-STAR.
C      LFB,LFA=INDICES OF STATIONS BELOW AND ABOVE T.E.
C      NCHB,NCHA=NUMBER OF CHANNELS BELOW AND ABOVE T.E.
C      LRF    = INDEX OF DUMMY ORTCHN LIST FOR THE T.E.
C      LRXF   = INDEX OF LAST CHANNEL BELOW THE T.E.
C      JORDER= 0 IF TOTAL FLOW AT X1F IS GIVEN
C            = 2 IF FLOW ABOVE T.E. IS GIVEN
C            = 1 IF FLOW BELOW T.E. IS GIVEN
C      JORDER= -1 IF FLOW AT X1F IS CHOKED AND SINGLE CHANNEL
      DIMENSION       X1F(1),X2F(1),X1BF(1),X1AF(1),
     1                S1F(1),NCHB(1),NCHA(1),JORDER(1),VNR(12)
        EQUIVALENCE   (LFB,X1BF),(LFA,X1AF),(LRF,NCHB),(LRXF,NCHA)
        DIMENSION     LFB(1),LFA(1),LRF(1),LRXF(1)
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
      DIMENSION       X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
     1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
     1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
     3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
        LOGICAL       PRIM
        INTEGER TYPELB,TYPEUB
        DIMENSION     SCHOKE(1)
        EQUIVALENCE   (SCHOKE,DWDV)
        EQUIVALENCE   (CHNAM,BDT,CH,X2W,X1F,X1)
        EQUIVALENCE   (LHNEXT,LBNEXT,LTNEXT,LWNEXT,X2F,LNEXT)
        EQUIVALENCE   (WTFLOW,LBZ1,NPT,S1W,X1BF,MLB)
        EQUIVALENCE   (TTO,CHNAME,LPSI,X1AF,MUB), (PTO,UP,LTT,S1F,PRIM)
        EQUIVALENCE   (TSO,LEDEX,LPT,NCHB,TYPELB)
        EQUIVALENCE   (PSO,ZBT,LRCU,NCHA,NAMELB)
       EQUIVALENCE   (MACHO,RBT,CRG,JORDER,ILB), (AO,ANGBT,CPGJ,VNR,FLB)
        EQUIVALENCE   (VARY,C2CP,S1LB), (RG,QGAM,TYPEUB)
```

```
      IF(GAMMA.NE.0.) ATINF=SQRT(GAMMA*CRG(LT)*TT(NSL))
      MINF   = MACHC
      UINF   = MACHC*SQRT(GAMMA*CRG(LT)*TSC)

  990 LTA    = LT
      RETURN
 1185 FORMAT(/1X20H***  ERROR- FOR CHN=A6,1X53HTHE STATIC PRESSURE EXCEE
     *DS THE INPUT TOTAL PRESSURE.,8H(BCONV) )
 1182 FORMAT(34H ***  ERROR- THE R (OR Y) FOR CHN=    ,A6,
     *35H MUST BE IN ASCENDING ORDER (BCONV)  )
 1183 FORMAT(21H ***  ERROR- FOR CHN=    ,A6,31H THE INPUT FLOW RATE OF
     *WTFLOW=,F9.3,37H IS GREATER THAN THE CHOKED VALUE OF ,F8.3,
     *8H (BCONV)  )
 1184 FORMAT(53H ***  ERROR- FAILURE OF PS-ITERATION GIVEN WTFLOW/CG=,
     *F9.4, 9H FOR CHN=   ,A6,8H (BCONV)  )
 1200 FORMAT(/1X32HERROR- THE FLOW RATE FOR CHANNEL2X,A6,1X15HIS NOT DEF
     *INED.)
      END
```

```
*DECK BLDTBS
      SUBROUTINE BLDTBS
*BLDTBS          BUILD ORTHOGONAL/CHANNEL TABLE,                -BLDTBS-
C                STREAMLINE TABLE, STATION TABLE,
C                FIELD TABLES AND FLOW ADJUSTMENT TABLE.

C     INPUT-
C      BOUNDARY TABLE, /BDYTAB/
C      CHANNEL INPUT DATE, /CHDATA/
C      ORDERED EDGE POINTS, /LETEPT/

C     OUTPUT-
C      LIST OF CHANNELS FOR EACH ORTHOGONAL, /ORTCHN/
C      TABLE OF CONVECTED PROPERTIES, /CONVTB/
C      STREAMLINE TABLE, /SLTAB/
C      STATION TABLE, /STATAB/
C      FIELD VALUES, /CZ/, /CR/, /CS2/, /CM/
C      TABLE OF STAS AT WHICH FLOW ADJUSTMENT MUST BE ACCOMPLISHED, /CADJ
C      TRAILING EDGE WAKE DISPLACEMENT THICKNESS TABLE, IF NOT CARD INPUT

       COMMON /ALLCOM/ MACHA,PSA,TSA,PTA,TTA, AXIA,RGA,GAMA,
      1                MACHC,PSC,TSC,PTC,TTC, AXIC,RGC,GAMC,
      2                DAXIT,SCALEA,TTE,CHOTST
         REAL          MACHA(1),MACHC
         LOGICAL       AXIA,AXIC,CHOTST
       COMMON /CBEAM2/ DR,DZ,YPA,YPB,F,G, DX,YQDX,ZM,RM,ANGM,CURVM,S1M,
      1                RZONLY, ANGCHD,SINTVL, YPASQ,YPAB,YPBSQ
         LOGICAL       RZONLY
C      INDEX- M=MO,NM
       COMMON /CVM    / VM(300)
       COMMON /CZ     / Z(300)
       COMMON /CR     / R(300)
       COMMON /CS2    / S2(300)
       COMMON /CS1    / S1(300)
       COMMON /CPHI1  / PHI1(300)
       COMMON /CM     / JMS(300)
       COMMON /CCURV  / CURV(300)
       COMMON /CB     / B(300)
       COMMON /CIDEX  / M,J,MU,MD,ISTAG
         DIMENSION    WSL(300)
         EQUIVALENCE  (WSL,CURV)
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
         DIMENSION    LIMITS(24)
         EQUIVALENCE  (LIMITS,LHO)
C     TABLE OF LEADING EDGE AND TRAILING EDGE POINTS
C      INDEX- LE=LEO,LEE,10
C      NLE,NTE=NO. OF L.E. AND T.E. COINCIDENT PTS, RESPECTIVELY
C      CHL,CHU=NAME OF CHANNEL ABOVE AND BELOW PT, RESPECTIVELY
C      BDL,BDU=BOUNDARY NAMES ASSOCIATED WITH THE POINTS
C      NUSED = COUNT OF TIMES THAT POINT USED IN CONSTRUCTION OF /ORTCHN/
       COMMON /LETEPT/ XE(1),YE(1),ANGE(1),NLE(1),NTE(1),
      1                CHL(1),CHU(1),BDL(1),BDU(1),NUSED(491)
         INTEGER       CHL,CHU,BDL,BDU
C     TABLE OF CHANNELS EMBRACED BY EACH ORTHOGONAL
C      INDEX- LR=LRO,LRE,LRD
```

```
      IF(AXIA) GO TO 170
      DO 165 J=1,NSL
  165 A(J)  = Y(J)
      GO TO 173
  170 DO 172 J=1,NSL
  172 A(J)  = PI*R(J)*R(J)
  173 PTMIN = PT(1)
      DO 174 J=2,NSL
      PTMIN = AMIN1(PTMIN,PT(J))
  174 IF(A(J).LT.A(J-1)) ERR=.TRUE.
      IF(ERR) GO TO 182

  175 QV    = 0.
  176 IF(PS.EQ.BITS) GO TO 177
      YTOL  = 1.E6
      GO TO 179
  177 PS(1) = .95*PTMIN
      YTOL  = WTF*1.E-5
  178 CALL SETM(1,PS, PS(2),NSL-1)
  179 DO 180 J=1,NSL
      TS    = TT(J)*(PS(J)/PT(J))**FGT(LT)
      IF(TS.GE.TT(J)) GO TO 185
      V(J)  = SQRT(C2CP(LT)*(TT(J)-TS))*PS(J)/(CRG(LT)*TS)
  180 CONTINUE
      PSI(1)= 0.
      CALL LSPFIT(A,V,NSL, A,PSI,NSL, -1)

      DELP  = PTMIN-PS(1)
      XJP   = .5*DELP
      DYDX  = -.5*PSI(NSL)/DELP
      YO    = WTF
      CALL QIREM(PS,PSI(NSL),XJP,QV)
      IF(QV.GE.2. .AND. QV(5).EQ.0.) GO TO 183
      IF(QV.EQ.21.) GO TO 184
      IF(QV.NE.0.) GO TO 178
C      *MACHC AND TSC FOR FAR FIELD CALCULATION (RARE OPTION)
       MACHC = V(NSL)/SQRT(GAMMA*CRG(LT)*TS)
       TSC   = TS
      GO TO 250

C     ERROR COMMENTS
  182 WRITE (6,1182) CHT
      GO TO 187
  183 PSIMAX= PSI(NSL)*CG
      WRITE (6,1183) CHT,WTFLOW(LH),PSIMAX
      GO TO 186
  184 WRITE (6,1184) WTF,CHT
      GO TO 186
  185 WRITE (6,1185) CHT
  186 CALL TABPRT(2HQV,QV,8,8)
      CALL TABPRT (6HCQIREM,YTOL,4,4)
      CALL TABPRT(3HPSI,PSI,NSL,10)
  187 ERR   = .TRUE.
      CALL TABPRT(2HPS,PS,NSL,10)
      CALL TABPRT(2HPT,PT,NSL,10)
      CALL TABPRT(2HTT,TT,NSL,10)
      CALL TABPRT(4HAREA,A,NSL,10)
      GO TO 250
```

```
C     GIVEN MACH NUMBER, AREA AND STATIC FLOW PROPERTIES
  190 IF(WTF.NE.0. .AND. (LH.EQ.0 .OR. MACHO(LH).EQ.BITS)) GO TO 200
      MACHC = MACHA
      IF(LH.EQ.0) GO TO 195
      IF(MACHO(LH).NE.BITS) MACHC=MACHO(LH)
      IF(PSO(LH).NE.BITS) PSC=PSO(LH)
      IF(TSO(LH).NE.BITS) TSC=TSO(LH)
  195 IF(GGAM(LT).EQ.0.) FG1=1./(TSC*CRG(LT))
      TTQTS = 1.+0.5*FG1*MACHC*MACHC
  196  IF(LH.EQ.0 .OR. (TTO(LH).EQ.BITS .AND. PTO(LH).EQ.BITS)) GOTO 197
C      *TOTAL CONDITIONS ARE SPECIFIED RATHER THAN STATIC
       TSC    = TT/TTQTS
       PSC    = PT/TTQTS**FGP(LT)
       GO TO 198
  197 TT     = TSC*TTQTS
      PT     = PSC*TTQTS**FGP(LT)
  198 IF(WTF.NE.0.) GO TO 240
      IF ( LH.NE. 0 .AND. AO(LH).NE.BITS ) AREA=AO(LH)*RHL
      IF ( LH.NE. 0 .AND. AO(LH).NE. BITS .AND. AXIA ) AREA=AO(LH)*PI
     * *RHL**2
      AREATB(LT)=AREA
      WTF    = PSC/(CRG(LT)*TSC)*AREA*MACHC
      IF(GGAM(LT).NE.0.) WTF=WTF*SQRT(GAMMA*CRG(LT)*TSC)
      GO TO 240

C     GIVEN FLOW RATE + TOTAL/STATIC CONDITIONS FROM STC/SHEET-1
  200 AREATB(LT)=0.
      IF(TSC.LT.TTC)
     *AREATB(LT)=WTF*CRG(LT)*TSC/(PSC*SQRT(C2CP(LT)*(TTC-TSC)))
  210 AREA   = AREATB(LT)

  240 PSI(NSL)=WTF
      IF(WTF.NE.0.) GO TO 250
      ERR    = .TRUE.
      WRITE (6,1200) CHT

C     PUT DATA IN CONVTB-ARRAY
  250 CH(LT)= CHT
      NPT(LT)=NSL
      LT1    = LT+15
      CALL MOVE(1, PSI,CH(LT1),NSL,1)
      LPSI(LT)=LT1-LT
      LT1    = LT1+NSL
      CALL MOVE(1, TT,CH(LT1),NSL,1)
      LTT(LT)=LT1-LT
      LT1    = LT1+NSL
      CALL MOVE(1, PT,CH(LT1),NSL,1)
      LPT(LT)=LT1-LT
      LT1    =LT1+NSL
      CALL MOVE(1, RCU,CH(LT1),NSL,1)
      LRCU(LT)=LT1-LT
      LTNEXT(LT)=15+4*NSL
      LTE    = LT+LTNEXT(LT)-1

C     EXTERNAL CHANNEL PROPERTIES FOR FAR FIELD CALC
      IF(CHT.NE.EXT) GO TO 990
      ATINF = 1.E6
```

```
      EQUIVALENCE   (MACHO,RBT,CRG,JORDER,ILB), (AO,ANGBT,CPGJ,VNR,FLB)
       EQUIVALENCE   (VARY,C2CP,SILB), (RG,QGAM,TYPEUB)
       EQUIVALENCE   (GAM,FGT,NAMEUB), (NR,FGP,IUB), (NC,FGR,FUB)
       EQUIVALENCE   (TAB(1),AREATB,SIUB), (TAB(2),VMB), (TAB(3),DWDV)
       EQUIVALENCE   (TAB(4),X2CL), (TAB(5),VCL), (TAB(6),MCL)
      COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
     *                LO,LESTA,  LDUM(8),
     *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
     *                LEO,LEE, LRO,LRE,LRD
       DIMENSION      LIMITS(24)
       EQUIVALENCE    (LIMITS,LHO)

      COMMON /CBITS / BITS,BLANK
      COMMON /CFRFIN/ ATINF,MINF,RFFREF,UINF,ZDN1,ZDN25
       REAL                 MINF
      COMMON /CGRAV / CG
      COMMON / CNORM / RHL,RM,AHL,ARM
      COMMON /CPI   / PI,TWOPI,PIQ2,PIQ4,TODEG,TORAD
      COMMON /CQIREM/ YTOL,YO,DYDX,CTRMAX
      COMMON /CTAPOS/ RESTRT,ENDBDT,ENDFIL,K6SV
       LOGICAL        RESTRT,ENDBDT,ENDFIL
      COMMON /ERASE/  EDUM(72),QV(8), A(90),V(90),
     1                PSI(90),R(90),TT(90),PT(90),RCU(90),PS(90)
       DIMENSION      Y(90)
       EQUIVALENCE    (Y,R)
      COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
       LOGICAL        ERR,ERRMAJ,INERR,PRERR

       INTEGER        CHT,EXT

      DATA EXT/3HEXT/

      CHT   = CHTA
      CALL SETM(1,BITS, PSI,540)
      CALL RTCFI(CHT,LH)

C     DEFINE GAS PROPERTIES
      LT    = LTE+1
      LTE   = LTE+15
      QGAM(LT)=0.
      FGR(LT)=0.
      FGP(LT)=1.
      FGT(LT)=1.
      GAMMA = GAMA
      IF(LH.NE.0 .AND. GAM(LH).NE.BITS) GAMMA=GAM(LH)
      IF(GAMMA.EQ.0.) GO TO 85
      FG1   = GAMMA-1.
      FGR(LT)=1./FG1
      FGP(LT)=GAMMA*FGR(LT)
      FGT(LT)=FG1/GAMMA
      QGAM(LT)=1./GAMMA

   85 CRG(LT)=RGA
      IF(LH.NE.0 .AND. RG(LH).NE.BITS) CRG(LT)=RG(LH)
      CPGJ(LT)=FGP(LT)*CRG(LT)
      C2CP(LT)=2.*CPGJ(LT)

C     DEFINE TOTAL PROPERTIES AS DETERMINED FROM DATA ON
```

```
C      STC/SHEET-1 OF INPUT
       TTC    = TTA
       PTC    = PTA
       PSC    = PSA
       IF(GAMA.EQ.0.) GO TO 95
       FG2    = GAMA-1.
       FGRA   = GAMA/FG2
       GO TO 97
    95 FG2    = 1./(TSA*RGA)
       FGRA   = 1.
    97 IF(MACHA.EQ.BITS) GO TO 99
       TTQTS  = 1.+.5*FG2*MACHA*MACHA
       TTC    = TSA*TTQTS
       PTC    = PSA*TTQTS**FGRA
       TSC    = TSA
       GO TO 100
    99 TSC    = TTC*(PSC/PTC)**(1./FGRA)
       TTQTS  = TTC/TSC
       MACHC  = SQRT(2.*(TTQTS-1.)/FG2)

C     NUMBER OF INPUT STREAMLINES.  GIVEN FLOW RATE.
   100 NSL    = 1
       WTF    = 0.
       IF(LH.EQ.0) GO TO 150
       NSL    = NR(LH)
       IF(WTFLOW(LH).NE.BITS) WTF=WTFLOW(LH)/CG

C     NO INPUT PROFILES
       IF(NSL.NE.0) GO TO 150
       TT     = TTC
       PT     = PTC
       IF(TTO(LH).NE.BITS) TT=TTO(LH)
       IF(PTO(LH).NE.BITS) PT=PTO(LH)
       NSL    = 1

C     FILL PS, PT, TT AND RCU TABLES
   150 IF(TT.EQ.BITS) TT=TTC
       IF(PT.EQ.BITS) PT=PTC
       IF(RCU.EQ.BITS) RCU=0.
       IF(PSI.EQ.BITS) GO TO 160
       CALL FILL(PSI,PT,1,NSL)
       CALL FILL(PSI,TT,1,NSL)
       CALL FILL(PSI,RCU,1,NSL)
       IF(WTF.EQ.0.) GO TO 250
       CONST = WTF/PSI(NSL)
       DO 155 J=1,NSL
   155 PSI(J)= CONST*PSI(J)
       GO TO 250

   160 IF(R.EQ.BITS) GO TO 190
       CALL FILL(R,PT,1,NSL)
       CALL FILL(R,TT,1,NSL)
       CALL FILL(R,RCU,1,NSL)
       IF(PS.NE.BITS) CALL FILL(R,PS,1,NSL)

C     INTEGRATION OF RHO*V*DA
       IF(NSL.EQ.1) GO TO 190
       IF(PS.EQ.BITS .AND. WTF.EQ.0.) CALL ERROR1
```

```
*DECK BUILDS
      PROGRAM BUILDS
      COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
     *                LO,LESTA,  LDUM(8),
     *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
     *                LEO,LEE, LRO,LRE,LRD
        DIMENSION     LIMITS(24)
        EQUIVALENCE   (LIMITS,LHO)
C     FLOW ADJUSTMENT TABLE
C      INDEX- LF=LFO,LFE
       COMMON /CHDATA/ X1F(1),X2F(1),X1BF(1),X1AF(1),
     1                 S1F(1),NCHB(1),NCHA(1),JORDER(1),VNR(12)
        EQUIVALENCE   (LFB,X1BF),(LFA,X1AF),(LRF,NCHB),(LRXF,NCHA)
        DIMENSION     LFB(1),LFA(1),LRF(1),LRXF(1)
        DIMENSION TABLES(10)
        EQUIVALENCE (TABLES(1),X1F(1))
       COMMON /CLWOSV/ LWOSV

       COMMON /SELECT/ LENTRY

       GO TO (5,10), LENTRY
     5 CALL BLDTBS
       GO TO 20
C     REBUILD CONVECTED PROPERTIES TABLE AND REPACK IF RESTRT=T.
    10 CALL RBCONV
       NMOVE1= LTE-LBDO+1
       LWTO  = LHE+1+NMOVE1
       CALL MOVE(2, TABLES(LBDO),TABLES(LHE+1),NMOVE1,1,
     1              TABLES(LWO),TABLES(LWTO),LESTA-LWOSV+1,1)
       MOVE1 = LHE+1-LBDO
       LBDO  = LBDO+MOVE1
       LBDE  = LBDE+MOVE1
       LTO   = LTO+MOVE1
       LTE   = LTE+MOVE1
       LWO   = LWTO
       MOVE2 = LWO-LWOSV
       LWE   = LWE+MOVE2
       LFO   = LFO+MOVE2
       LFE   = LFE+MOVE2
       LO    = LO+MOVE2
       LESTA = LESTA+MOVE2

C     SET FLOW ADJUSTMENT ITERATION COUNTER TO ZERO
       LF    = LFO
   850 IF(LF.GE.LFE) GO TO 20
       VNR(LF)= 0.
       LF    = LF+NFCOLS
       GO TO 850

    20 RETURN
       END
```

```
*DECK BCONV
      SUBROUTINE BCONV(CHTA,LTA,AREA)
*BCONV-         BUILD CONVECTED PROPERTIES TABLE                -BCONV-
       INTEGER          CHTA

C     INPUT-
C      CHTA   = CHANNEL NAME
C      AREA   = FLOW AREA IN CASE NO /CHDATA/ IS AVAILABLE
C      DATA IN THE CHANNEL DATA TABLE, /CHDATA/

C     OUTPUT-
C      LTA    = INDEX OF CHTA IN CONVTB
C      SUBTABLE OF CONVECTED FLOW PROPERTIES
C      DETERMINATION OF CHANNEL FLOW RATE

C     OUTPUT FOR FAR FIELD CALCULATION
C      ATINF  = SPEED OF SOUND AT STAGNATION TEMPERATURE
C      MINF   = FREE STREAM MACH NUMBER
C      UINF   = FREE STREAM VELOCITY

       COMMON /ALLCOM/ MACHA,PSA,TSA,PTA,TTA, AXIA,RGA,GAMA,
      1                MACHC,PSC,TSC,PTC,TTC, AXIC,RGC,GAMC,
      2                DAXIT,SCALEA,TTE,CHOTST
         REAL          MACHA(1),MACHC
         LOGICAL       AXIA,AXIC,CHOTST
C     CHANNEL INPUT DATA TABLE
C      INDEX- LH=LHO,LHE
C     TABLE OF CONVECTED PROPERTIES
C      INDEX- LT=LTO,LTE
C      CH     = CHANNELNAME
C      LTNEXT= INDEX INCREMENT TO THE NEXT CHANNEL
C      LPSI   = RELATIVE LOCATION OF PSI LIST
C      NPT    = NO. OF PSI, TT, PT AND RCU VALUES
C      LTT    = RELATIVE LOCATION OF TT LIST
C      LPT    = RELATIVE LOCATION OF PT LIST
C      LRCU   = RELATIVE LOCATION OF RCU LIST
       COMMON /CHDATA/ CHNAM(1),LHNEXT(1),WTFLOW(1),TTO(1),PTO(1),
      1                TSO(1),PSO(1),MACHO(1),AO(1),VARY(1),
      2                RG(1),GAM(1), NR(1),NC(1),TAB(6),
      4                BB(75)
         LOGICAL       VARY
         INTEGER CHNAM
         DIMENSION     VO(1)
         REAL          MACHO
         EQUIVALENCE   (VO,MACHO)
        DIMENSION      CH(1),LTNEXT(1),NPT(1),LPSI(1),LTT(1),LPT(1),
      1                LRCU(1),
      2                CRG(1),CPGJ(1),C2CP(1),QGAM(1),FGT(1),FGP(1),
      3                FGR(1),AREATB(485)
         INTEGER CH
        DIMENSION XCH(1)
        EQUIVALENCE (CH,XCH)
         EQUIVALENCE   (CHNAM,BDT,CH,X2W,X1F,X1)
         EQUIVALENCE   (LHNEXT,LBNEXT,LTNEXT,LWNEXT,X2F,LNEXT)
         EQUIVALENCE   (WTFLOW,LBZ1,NPT,S1W,X1BF,MLB)
         EQUIVALENCE   (TTO,CHNAME,LPSI,X1AF,MUB), (PTO,UP,LTT,S1F,PRIM)
         EQUIVALENCE   (TSO,LEDEX,LPI,NCHB,TYPELB)
         EQUIVALENCE   (PSO,ZBT,LRCU,NCHA,NAMELB)
```

```
C       X2W,LWNEXT(=2+2N), S1W(1),S1W(2)...S1W(N), DST(1),DST(2),..DST(N)
C      X2W    = STREAMLINE COORDINATE
C      S1W    = DISTANCE ALONG STREAMLINE FROM T.E.
C      DST    = WAKE DISPLACEMENT THICKNESS AS A FUNCTION OF S1W
       EQUIVALENCE    (CH,X1F,X2W), (LTNEXT,X2F,LWNEXT), (NPT,X1BF,S1W)
       EQUIVALENCE    (LPSI,X1AF), (LTT,S1F), (LPT,NCHB), (LRCU,NCHA)
       EQUIVALENCE    (CRG,JORDER), (CPGJ,VNR)

       COMMON /CIDEX / M,J,MU,MD,ISTAG
       COMMON /CMAXIT/ MAXIT,MAJCTR,GREFIN,EDUM
       COMMON /CPTMOV/ VELPOT,ICOB,NODENS,CPTDUM
       COMMON /CR    / R(300)
       COMMON /CS1   / S1(300)
       COMMON /CTHICK/ NTHKX,NTHKY,THKX(20),THKY(20),THIK2D(78)
       COMMON /CZ    / Z(300)
       COMMON /ERASE / PSI(800)

       INTEGER        CHX

C     INTERPOLATE FOR LAMINA THICKNESS
       NK     = MB-MA+1
       CALL SETM(1.,1., LAM,NK)
       IF(NTHKX.LE.1) GO TO 100
       CALL LFIT2D(Z(MA),R(MA),LAM,NK)

C     INITIALIZE
  100 WAKE   = .FALSE.

C     DEFINE NUMBER OF STREAMLINES,  NK,  ASSOCIATED WITH EACH CHANNEL
       K      = 1
       M      = MA
       WADD   = 0.
  105  NK     = 0
       K1     = K
       M1     = M
  110  CALL GETIX
       IF(M.NE.M1) GO TO 114
       CHX    = SLCHN(J)
       PSI1   = X2(J)
  114  IF(SLCHN(J).NE.CHX) GO TO 120
       NK     = NK+1
       DISP(K)=0.
       WSTA(K)=W(J)+WADD
       PSI(NK)=X2(J)
       K      = K+1
       M      = M+1
       IF(M.LE.MB) GO TO 110

C     FIND INDEX IN CONVTB
  120  LT     = LTO
  125  IF(LT.GT.LTE) CALL ERROR1
       IF(CH(LT).EQ.CHX) GO TO 130
       LT     = LT+LTNEXT(LT)
       GO TO 125

C     INTERPOLATE FOR CONVECTED PROPERTIES
C      SCALE THE PSI TABLE TO CONFORM TO THE LPSI-TABLE IN /CONVTB/
  130  NI     = NPT(LT)
```

```
      I      = LT+LPSI(LT)
      I2     = I+NI
      IF(K1.EQ.1 .AND. NK.EQ.1) PSI1=PSI1-8.
      PSI1   = 8.*AINT(PSI1/8.)
      F      = XCH(I2-1)/8.
      DO 140 KN=1,NK
  140 PSI(KN)=(PSI(KN)-PSI1)*F
      IT     = LT+LTT(LT)
      IP     = LT+LPT(LT)
      IS     = LT+LRCU(LT)
      CALL LSPFIT(CH(I),CH(IT),NI, PSI,TT(K1),NK, 0)
      CALL LSPFIT(CH(I),CH(IP),NI, PSI,PT(K1),NK, 0)
C     CALL LSPFIT(CH(I),CH(IS),NI, PSI,RCU(K1),NK, 0)
      CALL SETM(1,CRG(LT),RGX(K1),NK)
      CALL SETM(1,C2CP(LT),C2CPX(K1),NK)
      CALL SETM(1,FGR(LT),FGRX(K1),NK)

C     WAKE DISPLACEMENT THICKNESS
C      SEARCH FOR X2-SUBTABLE
      IF(M.GT.MB) GO TO 200
      X2J    = X2(J)
      DISP(K-1)=-1.
      LW     = LWO
  155 IF(LW.GE.LWE) GO TO 190
      IF(X2W(LW).EQ.X2J) GO TO 170
      LW     = LW+LWNEXT(LW)
      GO TO 155
C      FIND TRAILING EDGE S1 IN THE FLOW ADJUSTMENT TABLE, S1F
  170 LF     = LFO
  175 IF(X2F(LF).EQ.X2J) GO TO 180
      LF     = LF+NFCOLS
      IF(LF.LT.LFE) GO TO 175
      CALL ERROR1
C      INTERPOLATE FOR WAKE DISPLACEMENT THICKNESS, DSTAR
  180 S1FTE=S1(M)-S1F(LF)
C       S1-FROM-T.E.
      IF(S1FTE.LE.0.) GO TO 190
      N      = (LWNEXT(LW)-2)/2
      LSTAR  = LW+N
      CALL LSPFIT(S1W(LW),DST(LSTAR),N, S1FTE,DISP(K-1),1, 0)
      IF(DISP(K-1)) 184,184,186
  184 DISP(K-1)=-1.
      GO TO 190
  186 WAKE   = .TRUE.

C     LOOP FOR NEXT CHANNEL
  190 WADD   = WSTA(K-1)
      GO TO 105

C     USE CONSTANT DENSITY APPROXIMATION FOR  MAJCTR.LE.NODENS
  200 IF(MAJCTR.LE.NODENS) CALL SETM(1,0.,FGRX,K-1)
      RETURN
      END
      OVERLAY(STC,1,3)
```

```
*DECK MATINV
      SUBROUTINE MATINV(YIJ,N,UNIT,M,DET,IN1,IN2,ND,IF)
CMATINV         DUMMY (CDC) MATINV SIMULATOR                          -MATINV-
      DIMENSION YIJ(1),UNIT(1),IN1(1),IN2(1)
      NN     = N*N
      CALL SETM(1,0.,UNIT,NN)
      N1     = N+1
      DO 1 L=1,NN,N1
    1 UNIT(L)= 1.
      CALL LRMDS1(YIJ,N,IN1,IN2,DET,IF,N)
      IF( DET.EQ.0. ) CALL ERROR1
      CALL DBSRT1(UNIT,N,IN1,IN2,YIJ,N,N)
    2 RETURN
      END
```

```
*DECK DUP2
      SUBROUTINE TTPT(MA,MB, WSTA,DISP,WAKE,TT,PT,LAM,RGX,C2CPX,FGRX)
*TTPT--          TT, PT, AND RCU FOR STREAMLINES                -TTPT-
       LOGICAL                              WAKE
       REAL                                          LAM(25)
       DIMENSION WSTA(25),DISP(25),TT(25),PT(25),
     1           RGX(25),C2CPX(25),FGRX(25)


C     INPUT-
C      MA     = FIRST FIELD POINT
C      MB     = LAST FIELD POINT

C     OUTPUT-
C      WSTA   = LIST OF STREAM FUNCTION VALUES
C      DISP(K)=NON-ZERO FOR POSSIBLE SLIP CONDITION BETWEEN STREAMLINE
C                K AND K+1,  OTHERWISE DISP(K)=0.
C              = DISPLACEMENT THICKNESS OF WAKE IF POSITIVE
C      WAKE   = .TRUE. IF THERE EXISTS ANY WAKE DISPLACEMENTS.
C      TT     = INTERPOLATED TOTAL TEMPERATURE
C      PT     = INTERPOLATED TOTAL PRESSURE
C      LAMBDA= LAMINA THICKNESS IN THIRD DIMENSION,  BLOCKAGE EFFECT
C      RCU    = INTERPOLATED ANGULAR MOMENTUM          ***NOT NOW IN USE
C      RGX    = GAS CONSTANT
C      C2CPX  = SPECIFIC HEAT
C      FGRX   = 1./(GAM-1.)= FUNCTION OF GAMMA FOR CALCULATING DENSITY
C      NOTE - LENGTH OF WSTA,TT,PT,RCU-LISTS IS MB-MA+1

       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
         DIMENSION     LIMITS(24)
         EQUIVALENCE   (LIMITS,LHO)
       COMMON /SLTAB / W(128),X2(128),SLCHN(128)
         INTEGER SLCHN
C     TABLE OF CONVECTED PROPERTIES
C      INDEX- LT=LTO,LTE
       COMMON /CHDATA/ CH(1),LTNEXT(1),NPT(1),LPSI(1),LTT(1),LPT(1),
     1                 LRCU(1),
     2                 CRG(1),CPGJ(1),C2CP(1),QGAM(1),FGT(1),FGP(1),
     3                 FGR(1),AREATB(485)
         INTEGER CH
       DIMENSION XCH(1)
       EQUIVALENCE (CH,XCH)
*   SEE OTHER LISTING OF TTPT FOR EXPLANATION OF VARIABLES
C     FLOW ADJUSTMENT TABLE
C      INDEX- LF=LFO,LFE
         DIMENSION     X1F(1),X2F(1),X1BF(1),X1AF(1),
     1                 S1F(1),NCHB(1),NCHA(1),JORDER(1),VNR(12)
         EQUIVALENCE   (LFB,X1BF),(LFA,X1AF),(LRF,NCHB),(LRXF,NCHA)
         DIMENSION     LFB(1),LFA(1),LRF(1),LRXF(1)
C     TABLE OF WAKE DISPLACEMENT THICKNESS
C      INDEX- LW=LWO,LWE
         DIMENSION              X2W(1),LWNEXT(1),S1W(47)
         DIMENSION     DST(1)
         EQUIVALENCE   (DST,S1W)
C      SUBTABLE ARRANGEMENT IS-
```

```
*DECK DUP1
      SUBROUTINE LFIT2D(X,Y,TO,NXY)
*LFIT2D          LINEAR SURFACE INTERPOLATION                    -LFIT2D-
C                IN A RECTANGULAR GRID
      DIMENSION        X(2),Y(2),TO(2)

C     INPUT-
C      X,Y    = LIST OF COORDINATES AT WHICH INTERPOLATED VALUES ARE TO BE
C      NXY    = NO OF COORDINATE POINTS

C      NXT    = NUMBER OF XT
C      NYT    = NUMBER OF YT
C      XT     = X-GRID OF T-TABLE
C      YT     = Y-GRID OF T-TABLE
C      T      = TABLE OF VALUES
C      NOTE   - NUMBER OF T-VALUES IS NXT*NYT, ORDER IS ILLUSTRATED BELOW
C         YT(NYT)-  T(3)       T(6)       T(NXT*NYT)
C         YT(2)  -  T(2)       T(5)       T(8)
C         YT(1)  -  T(1)       T(4)       T(7)
C                  ----------------------------------
C                     XT(1)     XT(2)      XT(NXT)

C     OUTPUT-
C      TO     = INTERPOLATED VALUES AT X,Y

      COMMON /CTHICK/ NXT,NYT,XT(20),YT(20),T(78)
      COMMON /ERASE / DUM(400),T1(200),T2(200)

C     FIND CORRECT X-INTERVAL
      I      = 1
      M      = 1
      ISV    = 0
  100 NCOUNT= 0
  105 IF(X(M).LT.XT(I)) GO TO 110
      IF(X(M).GT.XT(I+1)) GO TO 120
      F      = (X(M)-XT(I))/(XT(I+1)-XT(I))
      GO TO 150
  110 IF(I.EQ.1) GO TO 140
      I      = I-1
      GO TO 125
  120 IF((I+1).GE.NXT) GO TO 145
      I      = I+1
  125 NCOUNT= NCOUNT+1
      IF(NCOUNT.GT.NXT) CALL ERROR1
      GO TO 105
  140 F      = 0.
      GO TO 150
  145 F      = 1.

C     INTERPOLATE WRT Y
  150 IF(I.EQ.ISV) GO TO 160
      IJ2    = I*NYT+1
      IJ1    = IJ2-NYT
      CALL LFIT1(YT,T(IJ1),NYT, Y,T1,NXY)
      CALL LFIT1(YT,T(IJ2),NYT, Y,T2,NXY)
      ISV    = I

C     INTERPOLATE WRT X
```

225

```
  160 TO(M) = F*T2(M)+(1.-F)*T1(M)

      M     = M+1
      IF(M.LE.NXY) GO TO 100
C...  END LOOP FOR INTERPOLATIONG TO(M) AT X(M),Y(M),M=1,NXY

      RETURN
      END
```

```
      LOGICAL          ONCE,SETAG

      DATA KLE/2HLE/, KTE/2HTE/, KFIELD/5HFIELD/
      DATA KFAR/6HFARFLD/, KFREE/4HFREE/, KPRES/4HPRES/, KSOLID/5HSOLID/
      DATA ONCE/.FALSE./

C     CHECK FOR INCORRECT CHANNEL INPUT NAMES
      LH     = LHO
      GO TO 45
C       LOOP THROUGH BOUNDARY TABLE TO SEE IF CHNAM(LH) IS REFERENCED
   32 LB     = LBDO
   35 IF(CHNAM(LH).EQ.CHNAME(LB) .OR. CHNAM(LH).EQ.CHNAME(LB+1))GO TO 40
      LB     = LB+LBNEXT(LB)
      IF(LB.LT.LBDE) GO TO 35
C       NO REFERENCE FOUND FOR CHNAM(LH)
      ERR    = .TRUE.
      WRITE (6,1035) CHNAM(LH)
 1035 FORMAT(57H ***  ERROR - BOUNDARY INPUT DATA DOES NOT REFERENCE CHN
     1=,A6)
C       INDEX TO NEXT CHANNEL
   40 LH     = LH+LHNEXT(LH)
   45 IF(LH.LT.LHE) GO TO 32

C     LOOP THROUGH STATION TABLE TO INSERT SPECIAL BOUNDARY TYPES
      L      = LO

C       LOWER BOUNDARY
  100 NAMB   = NAMELB(L)
      KTYPE  = TYPELB(L)
      ITVL   = ILB(L)
      IRET   = 0
      GO TO 500
  150 TYPELB(L)=KTYPE
      IF(KTYPE.NE.KSOLID) NAMELB(L)=NAMB

C       UPPER BOUNDARY
      NAMB   = NAMEUB(L)
      KTYPE  = TYPEUB(L)
      ITVL   = IUB(L)
      IRET   = 1
      GO TO 500
  250 TYPEUB(L)=KTYPE
      IF(KTYPE.NE.KSOLID) NAMEUB(L)=NAMB

C     INDEX TO NEXT STATION
      L      = L+LNEXT(L)
      IF(L.LT.LESTA) GO TO 100
      RETURN

C**   GENERAL LOGIC FOR EITHER UPPER OR LOWER BOUNDARY
C       NAMB   = BOUNDARY NAME
C       KTYPE  = BOUNDARY TYPE
  500 IF(KTYPE.EQ.KLE .OR. KTYPE.EQ.KTE .OR. KTYPE.EQ.KFIELD)
     * GO TO 599

C     CHECK BOUNDARY TABLE TO FIND SEGMENT NAME IF TYPE=SOLID.
      SETAG  = .FALSE.
      LB     = LBF(NAMB)
```

227

```
      NAMBD = NAMB
      IF(KTYPE.NE.KSOLID) GO TO 520
      LBXN  = LBZ1(LB)
      IF(LBXN.EQ.0) GO TO 520
      LBX   = LBZ1(LB)+3*ITVL-3
C     LBX   = INDEX (RELATIVE TO SUBTABLE ORIGIN) OF THE
C               INTERVAL OF THE OL-BOUNDARY INTERSECTION POINT
      LB1   = LB
  510 IF(LBA(LB1).LE.LBX .AND. LBB(LB1).GE.LBX) GO TO 515
      LB1   = LB1+3
      IF(LB1.LT.(LB+LBZ1(LB))) GO TO 510
      CALL ERROR1
C       CHECK FOR FIRST OF DOUBLE POINTS ON UPPER BOUNDARY
  515   IF(IRET.EQ.0 .OR. LBX.NE.LBB(LB1) .OR. (LB1+3).GE.(LB+LBZ1(LB))
     *    .OR. LBA(LB1+3).NE.(LBB(LB1)+3)) GO TO 518
C       CHANGE STATION-TABLE REFERENCE TO THE 2ND PT (1ST STREAMWISE PT)
      NAMBD = BDNAME(LB1)
        LB1   = LB1+3
      IUB(L)=IUB(L)+1
        SETAG = .TRUE.
  518 NAMB  = BDNAME(LB1)

C     DETERMINE IF GIVEN BOUNDARY NAME HAS BEEN SPECIFIED BY
C      USER INPUT AS A SPECIAL BOUNDARY TYPE
  520 IF(NAMB.EQ.FARFLD(1) .OR. NAMB.EQ.FARFLD(2)) KTYPE=KFAR
      IF(NAMB.EQ.FREE(1) .OR. NAMB.EQ.FREE(2)) KTYPE=KFREE
      IF(NAMB.EQ.PRES(1) .OR. NAMB.EQ.PRES(2)) KTYPE=KPRES

C     SET ISTAG EQUAL TO ZERO AT THE SOLID/FREE BREAK POINT
      IF(.NOT.SETAG .OR. (NAMBD.NE.FREE(1) .AND. NAMBD.NE.FREE(2) .AND.
     *   NAMBD.NE.PRES(1) .AND. NAMBD.NE.PRES(2))) GO TO 530
      M     = MUB(L)
      CALL GETIX
      ISTAG = 0
      CALL SAVIX

C     FAR-FIELD BOUNDARY GEOMETRIC DATA
  530 IF(KTYPE.NE.KFAR .OR. ONCE) GO TO 599
      LB1   = LB+LBZ1(LB)
      LB2   = LB+LBNEXT(LB)-9
      RFFREF= RBT(LB2)
      ZDN1  = ZBT(LB2)
      ZDN25 = ZBT(LB1)
      WRITE (6,1530) RFFREF,ZDN1,ZDN25,NAMB
 1530 FORMAT(//2X,41HTHE FAR FIELD INTERFACE BOUNDARY IS AT R=,F9.3,
     *11H BETWEEN Z=,F9.3,4H AND,F9.3,1H.,8H   (BDY=,A6,1H))

C     SET UP FAR FIELD SOLUTION MATRIX
      CALL FRFDNZ
      ONCE  = .TRUE.

C     RETURN
  599 IF(IRET) 150,150,250
      END
```

228

```
      NFF    = NF
C     PARABOLIC FIT AT END POINTS OF FARFIELD BOUNDARY
      RA     = RFF(1)
      ZA     = ZFF(1)
      ZASQ   = 1./(Z1-ZA)**2
      A1     = R1+(RA-R1)*Z1**2*ZASQ
      C1     = (RA-R1)*ZASQ
      B1     = -2.*C1*Z1
      RB     = RFF(NFF)
      ZB     = ZFF(NFF)
      ZASQ   = 1./(Z25-ZB)**2
      A25    = R25+(RB-R25)*Z25**2*ZASQ
      C25    = (RB-R25)*ZASQ
      B25    = -2.*C25*Z25
C     LOCATE ENDPOINT INDICES
      DO 200 K=1,25
      IF( ZDN(K).GE.ZA ) GO TO 201
  200 CONTINUE
  201 LU     = K-1
      DO 210 K=1,25
      IF( ZDN(K).GT.ZB ) GO TO 211
  210 CONTINUE
  211 LD     = K
C     INTERPOLATE POINTS IN STC SOLUTION TABLES
      NUM    = LD-LU+1
      IF( PRFF.NE.0. ) CALL LSPFIT(ZFF,RFF,NFF,ZDN(LU+1),RDN(LU+1),
     * NUM,0)
C
C     INTERPOLATE CO-ORDINATE DERIVATIVES ON FAR-FIELD BOUNDARY
C
      IF( PRFFI.NE.0. ) GO TO 4
      CALL LFITI(ZFF,PHIFF,NFF,ZDN(LU+1),DRDN(LU+1),NUM)
      GO TO 555
    4 CALL LSPFIT(ZFF,PHIFF,NFF,ZDN(LU+1),DRDN(LU+1),NUM,0)
C     FILL END POINTS OF ZDN,DRDN TABLES
  555 DO 556 K=1,LU
      RDN(K)= A1+B1*ZDN(K)+C1*ZDN(K)**2
  556 DRDN(K)= B1+2.*C1*ZDN(K)
      DO 557 K=LD,25
      RDN(K)= A25+B25*ZDN(K)+C25*ZDN(K)**2
  557 DRDN(K)= B25+2.*C25*ZDN(K)

C     ADJUST DERIVATIVE AT ZDN POINTS CLOSEST TO
C     UPSTREAM / DOWNSTREAM STC POINTS

      DZDN   = ZDN(2)-ZDN(1)
      DZA1   = ZA-ZDN(LU)
      DZA2   = ZDN(LU+1)-ZA
      LUC    = LU
      IF( DZA2.GT.DZA1 ) GO TO 558
      LUC    = LU+1
  558 AA     = (ZA-ZDN(LUC))/DZDN
      SP     = B1+2.*C1*ZDN(LUC)
      IF( PRFFI.NE.0. ) GO TO 560
      CALL LFITI(ZFF,PHIFF,NFF,ZDN(LUC),SB,1)
      GO TO 561
  560 CALL LSPFIT(ZFF,PHIFF,NFF,ZDN(LUC),SB,1,0)
  561 ASSIGN 562 TO LGO
```

229

```
 5622 DRDN(LUC)= SP*(.5+AA)+SB*(.5-AA)
      GO TO LGO , (562,5)
  562 DZA1 = ZB-ZDN(LD)
      DZA2 = ZDN(LD-1)-ZB
      LUC  = LD
      IF( ABS(DZA2) .GT. ABS(DZA1) ) GO TO 563
      LUC  = LD-1
  563 AA   = (ZDN(LUC)-ZB)/DZDN
      SP   = B25+2.*C25*ZDN(LUC)
      IF( PRFFI.NE.0. ) GO TO 565
      CALL LFIT1(ZFF,PHIFF,NFF,ZDN(LUC),SB,1)
      GO TO 566
  565 CALL LSPFIT(ZFF,PHIFF,NFF,ZDN(LUC),SB,1,0)
  566 ASSIGN 5 TO LGO
      GO TO 5622
C
C     CALCULATE VELOCITIES ON FAR FIELD BOUNDARY
C
    5 DO 10 I=1,25
      SUM  = 0.
      DO 9 J=1,25
    9 SUM  = SUM+ZIJ(I,J)*DRDN(J)
   10 UDN(I)= (1.+SUM)*UINF
      IF( PRFF.EQ.0. ) GO TO 20
      WRITE (6,14)
      WRITE (6,15) (I,ZDN(I),RDN(I),DRDN(I),UDN(I),I=1,25)
   14 FORMAT(//3X,1HI,10X,3HZDN,13X,3HRDN,13X,4HDRDN,12X,3HUDN//)
   15 FORMAT(2X,I2,F17.6,F16.6,1PE17.6,0PF15.6)
C
   20 RETURN
      END
```

```
*DECK INSTA
      SUBROUTINE INSTA(LNEW,LBASE,L3,DOWNB,MA,MB)
*INSTA-                INSERT A STATION                          -INSTA-
       LOGICAL                          DOWNB

C     INPUT-
C      LNEW   = LOCATION IN STATION-TABLE OF NEW STATION
C      LBASE  = LOCATION OF BASE STATION
C      L3     = LOCATION OF DOWNSTREAM (OR UPSTREAM) STATION
C      DOWNB  = T IF L3 IS AN UPSTREAM STA, OTHERWISE =F
C      MA,MB  = NEW STATION FILED POINT INDEX LIMITS
C      Z,R,PHI1 FIELD VALUES

C     OUTPUT-
C      LNEW   = STATION FOLLOWING NEW STATION

       COMMON /ALLCOM/ MACHA,PSA,TSA,PTA,TTA, AXIA,RGA,GAMA,
      1                MACHC,PSC,TSC,PTC,TTC, AXIC,RGC,GAMC,
      2                DAXIT,SCALEA,TTE,CHOTST
        REAL           MACHA(1),MACHC
        LOGICAL        AXIA,AXIC
        LOGICAL        CHOTST
       COMMON /CBEAM2/ DR,DZ,YPA,YPB,F,G, DX,YQDX,ZM,RM,ANGM,CURVM,S1M,
      1                RZONLY, ANGCHD,SINTVL, YPASQ,YPAB,YPBSQ
        LOGICAL        RZONLY
C      INDEX- M=MO,NM
       COMMON /CZ     / Z(300)
       COMMON /CR     / R(300)
       COMMON /CS2    / S2(300)
       COMMON /CS1    / S1(300)
       COMMON /CPHI1  / PHI1(300)
       COMMON /CM     / JMS(300)
       COMMON /CCURV  / CURV(300)
       COMMON /CB     / B(300)
       COMMON /CIDEX  / M,J,MU,MD,ISTAG
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
        DIMENSION      LIMITS(24)
        EQUIVALENCE    (LIMITS,LHO)
       COMMON /SLTAB / W(128),X2(128),SLCHN(128)
        INTEGER SLCHN
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
       COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
      1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
      1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
      3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
        LOGICAL        PRIM
        INTEGER        TYPELB,TYPEUB
        DIMENSION      SCHOKE(1)
        EQUIVALENCE    (SCHOKE,DWDV)

       COMMON /CATAN3/ DANG
```

```
      COMMON /CBDYPT/ ANGD,CURVD
      COMMON /CBITS / BITS,IBLANK
      COMMON /CMAXIT/ MAXIT,MAJCTR,GREFIN,EDUM
      COMMON /CPI   / PI,TWOPI,PIQ2,PIQ4,TODEG,TORAD
      COMMON /CPRINT/ PDUM1(3),PREFIN
      COMMON /CVM   / VM(300)
      COMMON /ERASE / ASL(800)

      INTEGER         BDYNAM,FARFLD,FREE,FIELD,PRES,SOLID
      LOGICAL         UPU,UPD

      DATA FARFLD/6HFARFLD/, FIELD/5HFIELD/, FREE/4HFREE/, PRES/4HPRES/,
     *     SOLID/5HSOLID/

C*** RELOCATE TO MAKE ROOM FOR THE NEW STATION
C     INITIALIZE NEW-STATION VALUE TO THE BASE-STATION VALUES
C      CORRECT THE STA-TABLE INDICIES- L-END, L-BASE, L-THREE, L-UPSTREAM
      LN    = LNEW
      NMOVE = LN-1 - LESTA
      LB    = LBASE
      CALL MOVE(2, X1(LN),X1(LN+20),NMOVE,D, X1(LB),X1(LN),20,1)
      LESTA = LESTA+20
      LT    = L3+20
      LU    = LB
      IF(.NOT.DOWNB) GO TO 60
      LB    = LB+20
      LT    = L3
      LU    = L3

C     UPDATE THE POINTERS TO THE FIELD-TABLE
   60 NPTS  = MB-MA+1
      LNEXT(LN)=20
      CALL STTOFI(LN,NPTS)

C*** DEFINE STATION-TABLE VALUES FOR THE NEW STATION
      X1(LN) = .5*(X1(LB)+X1(LT))
      MLB(LN)=MA
      MUB(LN)=MB
      PRIM(LN)=.FALSE.
      X2CL(LN)=BITS

C**   LOWER BOUNDARY STATION-TABLE VALUES
      M     = MA
      CALL GETIX
      MX    = MU
      IF(DOWNB) MX=MD
      LX    = LU
      CALL STAND(MX,LX,UPPER)
      IF(MX-MLB(LX)) 210,220,250
  210 CALL ERROR1

C     LOWER BOUNDARIES OF NEW AND BASE STATIONS ARE ON THE SAME SL
  220 IF(TYPELB(LB).EQ.FIELD) GO TO 250
      IF(TYPELB(LB).EQ.FARFLD) GO TO 260

C     FREE BOUNDARY
      IF(TYPELB(LB).NE.FREE .AND. TYPELB(LT).NE.FREE) GO TO 224
      TYPELB(LN)=FREE
```

```
      GO TO 260

C     PRESSURE BOUNDARY
  224 IF(TYPELB(LB).NE.PRES .AND. TYPELB(LT).NE.PRES) GO TO 230
      TYPELB(LN)=PRES
      GO TO 260

C     SOLID BOUNDARY
  230 TYPELB(LN)=SOLID
      BDYNAM= NAMELB(LX)
      NAMELB(LN)=BDYNAM
      ILB(LN)=ILB(LX)
      FLB(LN)=FLB(LX)
      S1LB(LN)=S1LB(LX)
      LD    = LU
      CALL STANO(MU,LU,UPU)
      CALL STANO(MD,LD,UPD)
      DS1   = .5*(BARCS(BDYNAM,ILB(LU),ILB(LD)) + S1LB(LD)-S1LB(LU))
      IF(UPU.OR.UPD) CALL ERROR1
      IF(DOWNB) DS1=-DS1
      CALL BDYPTM(BDYNAM,ILB(LN),Z(M),R(M),FLB(LN),S1LB(LN),DS1,GMA)
      IF(GMA.NE.0.) CALL ERROR1
      PHI1(M)=ANGD
      B(M)  = .5*(B(MU)+B(MD))
      VM(M) = .5*(VM(MU)+VM(MD))
      IF(VM(M).EQ.0.) VM(M)=VM(MU+1)
      GO TO 300

C     INFIELD BOUNDARY
  250 TYPELB(LN)=FIELD
      ISTAG =3
      CALL SAVIX
      NAMELB(LN)=IBLANK
  260 ILB(LN)=0
      FLB(LN)=BITS
      S1LB(LN)=BITS

C**   UPPER BOUNDARY STATION-TABLE VALUES
  300 M     = MB
      CALL GETIX
      MX    = MU
      IF(DOWNB) MX=MD
      CALL STANO(MX,LX,UPPER)
      IF(MUB(LX)-MX) 310,320,350
  310 CALL ERROR1

C     UPPER BOUNDARIES OF  NEW  AND  BASE  STATIONS ARE ON THE SAME SL
  320 IF(TYPEUB(LB).EQ.FIELD) GO TO 350
      IF(TYPEUB(LB).EQ.FARFLD) GO TO 360

C     FREE BOUNDARY
      LD    = LU
      CALL STANO(MU,LU,UPU)
      CALL STANO(MD,LD,UPD)
      IF (TYPEUB(LB).NE.FREE .AND. TYPEUB(LD).NE.FREE) GO TO 324
      TYPEUB(LN)=FREE
      GO TO 360
```

233

```
C     PRESSURE BOUNDARY
  324 IF (TYPEUB(LB).NE.PRES .AND. TYPEUB(LD).NE.PRES) GO TO 330
      TYPEUB(LN)=PRES
      GO TO 360

C     SOLID BOUNDARY
  330 TYPEUB(LN)=SOLID
      BDYNAM= NAMEUB(LX)
      NAMEUB(LN)=BDYNAM
      IUB(LN)=IUB(LX)
      FUB(LN)=FUB(LX)
      S1UB(LN)=S1UB(LX)
      LD    = LU
      CALL STAND(MU,LU,UPU)
      CALL STAND(MD,LD,UPD)
      IF(.NOT.UPU .OR. .NOT.UPD) CALL ERROR1
      DS1   = .5*(BARCS(BDYNAM,IUB(LD),IUB(LU)) + S1UB(LU)-S1UB(LD))
      IF(.NOT.DOWNB) DS1=-DS1
      CALL BDYPTM(BDYNAM,IUB(LN),Z(M),R(M),FUB(LN),S1UB(LN),DS1,GMA)
      IF(GMA.NE.0.) CALL ERROR1
      PHI1(M)= ANGD-PI
      B(M)  = .5*(B(MU)+B(MD))
      VM(M) = .5*(VM(MU)+VM(MD))
      IF(VM(M).EQ.0.) VM(M)=VM(MU-1)
      GO TO 400

C     INFIELD BOUNDARY
  350 TYPEUB(LN)=FIELD
      ISTAG = 3
      CALL SAVIX
      NAMEUB(LN)=IBLANK
  360 IUB(LN)=0
      FUB(LN)=BITS
      S1UB(LN)=BITS

C     DEFINE THE FIELD POINTS BY CUBIC POLYNOMIAL INTERPOLATION ON SL
  400 M     = MA
      RZONLY= .TRUE.
      IF(TYPELB(LN).EQ.SOLID) GO TO 420
  410 CALL GETIX
      DZ    = Z(MD)-Z(MU)
      DR    = R(MD)-R(MU)
      F     = .5
      G     = .5
      ANGCHD= ATAN3(DR,DZ,PHI1(MU))
      YPA   = PHI1(MU)-ANGCHD
      YPB   = PHI1(MD)-ANGCHD
          MSV   = M
          MUSV  = MU
          MDSV  = MD
          M     = MD
          CALL GETIX
          ISTAGD= ISTAG
          MD    = M
          M     = MSV
          MU    = MUSV
          IF(ISTAGD.EQ.1) YPB=-YPA
      RZONLY= .FALSE.
```

```
      CALL BFI
      Z(M)   = Z(MU)+ZM
      R(M)   = R(MU)+RM
      PHI1(M)=ANGCHD+ANGM
      VM(M)  = F*VM(MD)+G*VM(MU)
      B(M)   = F*B(MD)+G*B(MU)
C      CHECK FOR POINTS ON A SLIP LINE
      IF(M.EQ.MA .OR. W(J).NE.0.) GO TO 420
      Z(M)   = .5*(Z(M-1)+Z(M))
      M      = M-1
      CALL GETIX
      M      = MSV
      DZ     = .25*(Z(MUSV)-Z(MU)+Z(MDSV)-Z(MD))
      DR     = .25*(R(MUSV)-R(MU)+R(MDSV)-R(MD))
      Z(M-1)= Z(M)-DZ
      R(M-1)= R(M)-DR
      Z(M)   = Z(M)+DZ
      R(M)   = R(M)+DR
  420 M      = M+1
      IF(M-MB) 410,425,500
  425 IF(TYPEUB(LN).NE.SOLID) GO TO 410

C     CHECK FOR OUT-OF-ORDER POINTS
  500 NORDER= 0
  502 NORDER= NORDER+1
      IF(NORDER.GE.20) CALL ERROR1
      MX1    = 0
      MAP1   = MA+1
      MSV    = MA
      S2(MA)= 0.
      DO 520 M=MAP1,MB
      DR     = R(M)-R(M-1)
      DZ     = Z(M)-Z(M-1)
      S2(M) = S2(M-1)+SQRT(DR*DR+DZ*DZ)
      CALL GETIX
      IF(W(J).EQ.0.) GO TO 518
      ANG2   = ATAN3(DR,DZ,PHI1(M-1))
      ADANG  = ABS(DANG-PIQ2)
      IF(MX1.NE.0) GO TO 515
      IF(ADANG.GE.PIQ2) MX1=MSV
      MSV    = M-1
  515 IF(ADANG.GE.PIQ2) MX2=M
      GO TO 520
  518 IF((M-1).EQ.MX2) MX2=M
  520 CONTINUE
C     DEFINE THE FIELD PT LOCATIONS BY UPSTREAM AREA DISTRIBUTIONS
      IF(MX1.EQ.0) GO TO 999
      MX1    = MAX0(MX1-NORDER,MA)
      MX2    = MIN0(MX2+NORDER,MB)
      WRITE (6,1550) MX1,MX2
 1550 FORMAT(14H INSTA-MX1,MX2,2I6)
      MX1    = MAX0(MX1-1,MA)
      MX2    = MIN0(MX2+1,MB)
C      ADD UP UPSTREAM AREAS
      M      = MX1
      CALL GETIX
      K      = 1
      ASL(1)= 0.
```

```
  562 MUM1   = MU
      M      = M+1
      K      = K+1
      CALL GETIX
      AREA   = SQRT((R(MU)-R(MUM1))*(R(MU)-R(MUM1)) +
     1              (Z(MU)-Z(MUM1))*(Z(MU)-Z(MUM1)))
      IF(AXIA) AREA=(R(MU)+R(MUM1))*AREA
      ASL(K)= ASL(K-1)+AREA
      IF(M.LT.MX2) GO TO 562
      ASLNK = ASL(K)

C       INTERPOLATE FOR COORDINATES
      DZBA   = Z(MX2)-Z(MX1)
      DRBA   = R(MX2)-R(MX1)
      DRSQBA= DRBA*(R(MX2)+R(MX1))
      RMASQ  = R(MX1)*R(MX1)
      DVMBA  = VM(MX2)-VM(MX1)
      M      = MX1+1
      K      = 2
  564 F      = ASL(K)/ASLNK
      Z(M)   = Z(MX1)+F*DZBA
      R(M)   = R(MX1)+F*DRBA
      IF(AXIA) R(M)=SQRT(RMASQ+F*DRSQBA)
      VM(M)  = VM(MX1)+F*DVMBA
      M      = M+1
      K      = K+1
      IF(M.LT.MX2) GO TO 564
      GO TO 502

  999 LNEW   = LN+20
      RETURN
      END
```

```
*DECK PTMOVE
      SUBROUTINE PTMOVE
*PTMOVE         POINT MOVEMENT ALONG STREAMLINES              -PTMOVE-

C     POINT MOVEMENT ALONG STREAMLINES TO OBTAIN AN ORTHOGONAL GRID


C     INPUT-
C      R,Z    = COORDINATES
C      PHI1   = ANGLE OF THE STREAMLINES
C      S1     = DISTANCES ALONG THE STREAMLINES

C     OUTPUT-
C      S2     = DISTANCES ALONG THE ORTHOGONALS
C      R,Z    = ADJUSTED COORDINATES
C      PHI1   = STREAMLINE ANGLES (ADJUSTED POINTS)
C      S1     = DISTANCES ALONG THE STREAMLINES (ADJUSTED)

       COMMON /CBEAM2/ DR,DZ,YPA,YPB,F,G, DX,YQDX,ZM,RM,ANGM,CURVM,S1M,
      1                RZONLY, ANGCHD,SINTVL, YPASQ,YPAB,YPBSQ
        LOGICAL        RZONLY
C      INDEX- M=MO,NM
       COMMON /CZ     / Z(300)
       COMMON /CR     / R(300)
       COMMON /CS2    / S2(300)
       COMMON /CS1    / S1(300)
       COMMON /CPHI1  / PHI1(300)
       COMMON /CM     / JMS(300)
       COMMON /CCURV  / CURV(300)
       COMMON /CB     / B(300)
       COMMON /CIDEX  / M,J,MU,MD,ISTAG
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
        DIMENSION      LIMITS(24)
        EQUIVALENCE    (LIMITS,LHO)
       COMMON /CBEND  / NBCB(2),ANGE(2),CURVE(2),FB(2)
       COMMON /CBITS  / BITS,BLANK
       COMMON /CCUBE  / NBC(2),C1(2),C2(2),FEND(2)
       COMMON /CGRAV  / CG
       COMMON /CPI    / PI,TWOPI,PIQ2,PIQ4,TODEG,TORAD
       COMMON /CREFIN/ SG1,SG2,VMG1,VMG2
      1,               NGR,NGZ, SGR(10),GR(10), SGZ(10),GZ(10)
       COMMON /SLTAB  / W(128),X2(128),SLCHN(128)
        INTEGER SLCHN
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
       COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
      1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
      1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
      3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
        LOGICAL        PRIM
        INTEGER TYPELB,TYPEUB
        DIMENSION      SCHOKE(1)
```

237

```
      EQUIVALENCE     (SCHOKE,DWDV)

      COMMON /CBDYPT/ ANGD,CURVD
      COMMON /CFB    / L,MA,MB,LX,IK,IKDIR,IKA,IKB,
     1                 NK,K,ADS1,XCHOKE,ADS1LB,ADS1UB,GMALB,GMAUB,
     2                 CFBDUM(17)
      COMMON /CINNER/ INRCTR,RDUM,NINNER(16),CNVF(16)
      COMMON /CMAXIT/ MAXIT,MAJCTR,GREFIN,EDUM
      COMMON /CPTMOV/ VELPOT,ICOB,CPTDUM(2)
       LOGICAL        VELPOT
      COMMON /CTOLRL/ TOLRL,MAXSWP,CLEN,DS2MX,TOLDS2,NSWP,
     1                 DS1DMP,DS1MXA,DS1MXB,DS1RMS,ES2MX,DS1RMO
      COMMON /CVM    / VM(300)
      COMMON /ERASE2/ X1L(128),SC(128),VC(128),VDS(128),FVDS(128),
     1                 SCX(128), PHI2(96),DS1(96),ZK(96),RK(96),WEZPT(96)
       LOGICAL        WEZPT
      COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
       LOGICAL        ERR,ERRMAJ,INERR,PRERR

      INTEGER         FIELD,SOLID,TE
      LOGICAL         OLWNCL

      DATA FIELD/5HFIELD/, NOMCL/6HNO MCL/, SOLID/5HSOLID/, TE/2HTE/
      DATA LE/2HLE/


C     INITAILIZE
      COEF   = 0.
      IF(DS1RMO.NE.0.) COEF=-DS1DMP/DS1RMO
      DS1MXA= 0.
      DS1MXB= 0.
      NDS1   = 0
      SDS1SQ= 0.

C     USE PARABOLIC END CONDITIONS ON THE ORTHOGONAL SPLINE FIT
      NBCB(1)=0
      NBCB(2)=0
      FB(1) = 0.
      FB(2) = 0.

C****CALCULATE POINT MOVEMENT ALONG CONTROL STREAMLINE

C     BUILD ARRAYS OF ARC DISTANCE AND VELOCITY
C      BY LOOPING THROUGH THE STATION-TABLE
      L      = LO
      LAST   = 0

C     FIRST POINT ON CONTROL STREAMLINE
  210 IF(L.GE.LESTA) GO TO 900
      IC     = 1
      LSAV   = L
      OLWNCL= .FALSE.
C      OLWNCL= ORTHOGONAL LINE WITH NO CONTROL SL, T OR F
      SC(1) = BITS
      X1A    = X1(L)
      XCNTRL= X2CL(L)

  220 X1L(IC)=X1(L)
```

238

```
      IF(SC(1).NE.BITS) GO TO 240
      MA     = MLB(L)
      MB     = MUB(L)
      DO 230 M=MA,MB
      CALL GETIX
      IF(X2(J)-XCNTRL) 230,232,230
  230 CONTINUE
      IF(IC.EQ.1) GO TO 245
      GO TO 243
  232 IF(IC.EQ.1) GO TO 240
C     (THE UPSTREAM OL OF THE REGION IS AT A T.E. IT DOES NOT INCLUDE AL
      X1L(1)= X1A
      SC(1) = S1(MU)
      VC(1) = VM(MU)
      IF(.NOT.VELPOT) VC(1)=500.

  240 SC(IC)=S1(M)
      SC(IC)= S1(M)
      VC(IC)= VM(M)
      IF(.NOT.VELPOT) VC(IC)=500.
      MCL(L)= M
C      IS CONTROL SL INCLUDED IN THE STATION STREAMLINES
       IF(M.LT.MLB(L)) CALL ERROR1
       IF(M.LE.MUB(L)) GO TO 244
C      CONTROL SL DOES NOT CROSS THIS OL, CHECK FOR FIELD BOUNDARIES
  243 IF(TYPELB(L).NE.FIELD .AND. TYPEUB(L).NE.FIELD) CALL ERROR1
       OLWNCL= .TRUE.
       MCL(L)= NOMCL
       GO TO 245
  244 M      = MD
      CALL GETIX

C    INDEX TO THE NEXT STATION
  245 IF(PRIM(L) .AND. IC.NE.1) GO TO 250
      L      = L+LNEXT(L)
      IC     = IC+1
      GO TO 220

C    LAST POINT ALONG CONTROL STREAMLINE
  250 X1B    = X1(L)

C    MODIFY STAGNATION POINT VELOCITY TO OBTAIN SMOOTH CURVE
C     ASSUME VELOCITY IS CONSTANT TO WITHIN 2 L.E. RADII
      IF(VC(1).NE.0.) GO TO 254
      M      = MCL(LSAV)
      SC21   = SC(2)-SC(1)
      F      = 2./(SC21*AMAX1(CURV(M),1.E-6))
      VC(1) = VC(2)*AMAX1(0.,1.-F)
  254 IF(VC(IC).NE.0.) GO TO 258
      M      = MCL(L)
      SC21   = SC(IC)-SC(IC-1)
      F      = 2./(SC21*AMAX1(CURV(M),1E-6))
      VC(IC)= VC(IC-1)*AMAX1(0.,1.-F)

C    INTEGRATION OF VC*DSC (I.E. CALC OF POTENTIAL FUNCTION)
  258 VDS(1)= 0.
      CALL LSUM(SC,VC,IC, VDS)
```

```
C     INTERPOLATION FOR ORTHOGONAL POSITIONS
      CONST = VDS(IC)/(X1B-X1A)
      NIC   = IC
      DO 260 IC=1,NIC
  260 FVDS(IC)=(X1L(IC)-X1A)*CONST
      CALL LSPFIT(VDS,SC,NIC, FVDS,SCX,NIC, 0)

C     LOOP THROUGH THE SAME STATIONS, ONCE FOR REGULAR OL-S, AGAIN FOR OL
      LOOP  = 1
C****CALCULATE ANGLE AND ARC LENGTH ALONG THE ORTHOGONALS
  300 L     = LSAV
      IC    = 1
      RZONLY= .FALSE.
C     LAST  = LAST ORTHOGONAL OF THE PREVIOUS REGION. (ALREADY ORTHOGONA
      IF(L.EQ.LAST) GO TO 450
  302 IF(LOOP.EQ.1 .AND. MCL(L).EQ.NOMCL) GO TO 450
      IF(LOOP.EQ.2 .AND. MCL(L).NE.NOMCL) GO TO 450
      MA    = MLB(L)
      MB    = MUB(L)
      NK    = MB-MA+1
C      RELOCATE Z,R TO ALLOW FOR DOUBLE SL-S
      M     = MA
      K     = 1
  308 ZK(K) = Z(M)
      RK(K) = R(M)
      WEZPT(K)=.FALSE.
      CALL GETIX
      IF(W(J).NE.0. .OR. K.EQ.1) GO TO 310
      WEZPT(K-1)=.TRUE.
      GO TO 312
  310 K     = K+1
  312 M     = M+1
      IF(M.LE.MB) GO TO 308
      NKX   = K-1

      PHI2(1)=PHI1(MA)+PIQ2
      S2(MA)= 0.
      CALL BFAS(ZK,RK,PHI2,S2(MA), 1,NKX)

C      LOCATE BACK  PHI2  AND  S2  IF DOUBLE SL OCCURED
      IF(NKX.EQ.NK) GO TO 322
      K     = NKX
  316 IF(.NOT.WEZPT(K)) GO TO 318
      M     = K-1+MA
      NMOVE = -(NKX-K+1)
      CALL MOVE(3, PHI2(K),PHI2(K+1),NMOVE,1, S2(M),S2(M+1),NMOVE,1,
     1             WEZPT(K),WEZPT(K+1),NMOVE,1)
      NKX   = NKX+1
  318 K     = K-1
      IF(K.GE.1) GO TO 316
      IF(NKX.NE.NK) CALL ERROR1

C     (BOUNDARY S1-TOLERANCE)
  322 TOLS1 = .02*S2(MB)/FLOAT(NK)

C     COMPUTE DEVIATION FROM  90 DEG  BETWEEN  STREAMLINES AND ORTHOGONAL
      K     = 1
      M     = MA
```

```
  325 PHI2(K)=PHI2(K)-(PHI1(M)+PIQ2)
      K      = K+1
      M      = M+1
      IF(M-MB) 325,325,328

C     CALCULATE POINT MOVEMENT ALONG STREAMLINES REQD FOR ORTHOGONALITY
  328 DS1(1)= 0.
      CALL LSPFIT(S2(MA),PHI2,NK, S2(MA),DS1,NK,-1)
C        CORRECT POSSIBLE JOG AT DOUBLE STREAMLINE
         K      = 1
         M      = MA
 3292    IF(.NOT.WEZPT(K)) GO TO 3294
         K      = K+1
         M      = M+1
         IF(.NOT.WEZPT(K)) CALL ERROR1
         DZ     = Z(M)-Z(M-1)
         DR     = R(M)-R(M-1)
         CS     = COS(PHI1(M))
         SN     = SIN(PHI1(M))
         S2MMM1= DR*CS-DZ*SN
         IF(S2MMM1.GE.0.) GO TO 3293
         Z(M)   = Z(M-1)
         R(M)   = R(M-1)
         S2(M) = S2(M-1)
         DS1(K)= DS1(K-1)
         GO TO 3294
 3293    DS1(K)= DS1(K-1) - DZ*CS-DR*SN
         S2(M) = S2(M-1) + S2MMM1
 3294    K      = K+1
         M      = M+1
         IF(M.LE.MB) GO TO 3292
      IF(LOOP.EQ.2) GO TO 3300
      K      = MCL(L)-MA +1
      ADS1   = SCX(IC)-SC(IC)-DS1(K)
      IF(TYPELB(L).EQ.LE) ADS1=0.
      IF(TYPEUB(L).EQ.LL) ADS1=-DS1(NK)
      GO TO 3314

C     PARTIAL OL WITH NO MCL, USE MIDDLE SL, EVAL. PT. MOVEMENT
 3300 MSV    = (MLB(L)+MUB(L))/2
      M      = MSV
      IK     = 65
      IKDIR = -1
      LX     = L
 3302 CALL STAND(M,LX,UPPER)
      X1L(IK)=X1(LX)
      SC(IK)= S1(M)
      VC(IK)= VM(M)
      IF(MCL(LX).NE.NOMCL) GO TO 3310
 3304 CALL GETIX
      IK     = IK+IKDIR
      IF(IKDIR) 3306,3308,3308
 3306 IKA    = IK
      IF(IK.LE.0) CALL ERROR1
      M      = MU
      GO TO 3302
 3308 IKB    = IK
      IF(IK.GT.128) CALL ERROR1
```

```
      M      = MD
      GO TO 3302
 3310 IF(IKDIR.EQ.1) GO TO 3312
      IKDIR = 1
      M      = MSV
      IK     = 65
      GO TO 3304
 3312 NIK    = IKB-IKA+1
      VDS(IKA)=0.
      CALL LSUM(SC(IKA),VC(IKA),NIK, VDS(IKA))
      FVDSK = (X1(L)-X1L(IKA))/(X1L(IKB)-X1L(IKA)) * VDS(IKB)
      CALL LSPFIT(VDS(IKA),SC(IKA),NIK, FVDSK,SCK,1, 0)
      K      = MSV-MA+1
      ADS1   = SCK-SC(65)-DS1(K)
C..   END SECTION FOR PARTIAL OL WITH NO MCL

C     CHECK TO SEE IF MAGNITUDE OF  DS1  IS REASONABLE
 3314 IF(ABS(DS1(NK)).LT.(.5*S2(MB))) GO TO 3316
      WRITE (6,1330) X1(L),L,MAJCTR
      IF(MAJCTR.LE.1) GO TO 3316
      WRITE (6,1331)
      ERR    = .TRUE.
      RETURN

C     DAMP THE CALCULATED STREAMWISE POINT SHIFTS
 3316 DO 3318 K=1,NK
      DS1(K)= DS1(K)+ADS1
      ABSDS1= ABS(DS1(K))
      DS1MXB= AMAX1(DS1MXB,ABSDS1)
      SDS1SQ= SDS1SQ + ABSDS1*ABSDS1
      NDS1   = NDS1+1
      DS1(K)= DS1(K)*EXP(COEF*ABSDS1)
 3318 DS1MXA= AMAX1(DS1MXA,ABS(DS1(K)))
      ADS1   = 0.

C     LOWER AND UPPER BOUNDARY POINT MOVEMENT
      ADS1LB= DS1(1)
      ADS1UB= -DS1(NK)

C     MOVE THE LOWER BOUNDARY POINT
      K      = 1
  332 GMALB = 0.
      GMAUB = 0.
      M      = MLB(L)
      CALL GETIX
      IF(TYPELB(L).NE.TE) GO TO 3332
      ADS1LB= 0.
      GO TO 3324
 3332 IF(TYPELB(L).NE.SOLID .OR. ISTAG.EQ.1) GO TO 334
 3324 MA     = MLB(L)
      IF(ADS1LB) 3325,3325,3326
 3325 IF(MU.NE.0) ADS1LB=AMAX1(ADS1LB,.5*(S1(MU)-S1(M)))
      GO TO 3327
 3326 IF(MD.NE.0) ADS1LB=AMIN1(ADS1LB,.5*(S1(MD)-S1(M)))
 3327 CALL BDYPTM(NAMELB(L),ILB(L),Z(MA),R(MA),FLB(L),S1LB(L),
     1             ADS1LB,GMALB)
      S1(MA)= S1(MA)+ADS1LB+GMALB
C       JUMP OVER RELOCATION OF ANGLE/CURVATURE IF -ITERATION FORMULA ON
```

```
C       BOUNDARY- (ICOB) IS LESS THAN OR EQUAL TO  MAJCTR.
        IF(MAJCTR.LE.ICOB) GO TO 333
        PHI1(MA)=ANGD
        CURV(MA)=CURVD
  333 MA     = MA+1
        K      = 2

C     MOVE THE UPPER BOUNDARY POINT
  334 M      =MUB(L)
        CALL GETIX
        IF(TYPEUB(L).NE.TE) GO TO 335
        ADS1UB= 0.
        GO TO 3352
  335 IF(TYPEUB(L).NE.SOLID .OR. ISTAG.EQ.1) GO TO 338
 3352 MB     = MUB(L)
        IF(ADS1UB) 3355,3355,3356
 3355 IF(MD.NE.0) ADS1UB=AMAX1(ADS1UB,.5*(S1(M)-S1(MD)))
        GO TO 3357
 3356 IF(MU.NE.0) ADS1UB=AMIN1(ADS1UB,.5*(S1(M)-S1(MU)))
 3357 CALL BDYPTM(NAMEUB(L),IUB(L),Z(MB),R(MB),FUB(L),S1UB(L),
     1             ADS1UB,GMAUB)
        S1(MB)= S1(MB)-ADS1UB-GMAUB
        IF(MAJCTR.LE.ICOB) GO TO 336
        PHI1(MB)=ANGD-PI
        CURV(MB)=-CURVD
  336 MB     = MB-1

C     CHECK FOR NON PRIM STATIONS EXTENDING BEYOUND THE ENDS OF THE BOUND
  338 IF(PRIM(L)) GO TO 340
        IF((GMALB+GMAUB).NE.0.) CALL ERROR1
        GO TO 348

C     PRIM STATIONS. IF EITHER -GET MINUS ASK- VALUE IS LARGE
C       CORRECT OTHER BOUNDARY.
  340 IF(IC.NE.1) GO TO 342
C       (FIRST STATION OF THE REGION)
        GMA    = AMAX1(GMALB,-GMAUB)
        GO TO 345
C       (LAST STATION OF THE REGION)
  342 GMA    = AMIN1(GMALB,-GMAUB)

  345 ADS1   = ADS1+GMA
        ADS1LB= -GMAUB
        ADS1UB= -GMALB
        IF(ABS(GMA).GE.TOLS1) GO TO 332
C       THIS IS A DANGEROUS LOOP

C     MOVE THE INTERIOR POINTS
  348 M      = MA
  350 CALL GETIX
        DS1(K)= DS1(K)+ADS1
        IF(DS1(K)) 360,400,380
C       (MOVE POINT UPSTREAM)
  360 IF(MU) 361,381,361
  361 DS1(K)= AMAX1(DS1(K),-.5*(S1(M)-S1(MU)))
        G      = -DS1(K)/(S1(M)-S1(MU))
        F      = 1.-G
        FB     = -G
```

243

```
      DR      = R(M)-R(MU)
      DZ      = Z(M)-Z(MU)
      PHIA    = PHI1(MU)
      PHIB    = PHI1(M)
      CURV(M)=CURV(MU)*G + CURV(M)*F
      GO TO 390
C     (MOVE POINT DOWNSTREAM)
  380 IF(MD) 381,361,381
  381 DS1(K)= AMIN1(DS1(K),.5*(S1(MD)-S1(M)))
C        CHECK FOR DOWNSTREAM LEADING EDGE STAGNATION POINT
C        THIS LOGIC VALID IF THERE ARE 2 OR MORE OL-S UPSTREAM OF STAG P
         MSV    = M
         M      = MD
         CALL GETIX
         ISTAGD= ISTAG
         M      = MSV
         CALL GETIX
C         CHECK FOR JUST ONE PT UPSTREAM OF STAG PT
         IF(ISTAGD.NE.1) GO TO 383
         IF(MU.EQ.0) CALL ERROR1
         GO TO 361
  383 F      = DS1(K)/(S1(MD)-S1(M))
      G      = 1.-F
      FB     = F
      DR     = R(MD)-R(M)
      DZ     = Z(MD)-Z(M)
      PHIA   = PHI1(M)
      PHIB   = PHI1(MD)
      CURV(M)=CURV(M)*G + CURV(MD)*F
  390 ANGCHD= ATAN3(DR,DZ,PHIA)
      YPA    = PHIA-ANGCHD
      YPB    = PHIB-ANGCHD
C     CALL BFI
       YQDX  = F*G*(G*YPA-F*YPB)
       ANGM  = YPA*(3.*G-2.)*G + YPB*(3.*F-2.)*F
      R(M)   = R(M) + (FB*DR+YQDX*DZ)
      Z(M)   = Z(M) + (FB*DZ-YQDX*DR)
      PHI1(M)=ANGCHD+ANGM
      S1(M)  = S1(M)+DS1(K)

  400 M      = M+1
      K      = K+1
      IF(M-MB) 350,350,450

C     INDEX TO THE NEXT STATION
  450 IF(IC.GE.NIC) GO TO 470
      L      = L+LNEXT(L)
      IC     = IC+1
      GO TO 302

C     LOOP AGAIN THROUGH STATIONS IF THERE ARE PARTIAL OL-S WITH NO MCL-S
  470 IF(.NOT.OLWNCL .OR. LOOP.EQ.2) GO TO 500
      LOOP   = 2
      GO TO 300

C     CONTINUE TO NEXT REGION
  500 LAST   = L
      IF(X2CL(L).EQ.BITS) L=L+LNEXT(L)
```

```
      GO TO 210

C     RMS OF THE REQUESTED DS1-S
  900 DS1RMS= SQRT(SDS1SQ/FLOAT(NDS1))
      IF(INRCTR.EQ.0) DS1RMO=DS1RMS
      RETURN


 1330 FORMAT(/1X44H***  THE ORTHOGONAL LINE ADJUSTMENTS AT STA=F6.3,1X3H
     *(L=13,34H) ARE UNREASONABLY LARGE.  MAJCTR=13,11H.  (PTMOVE))
 1331 FORMAT(6X40HPLEASE CHECK INPUT BOUNDARY COORDINATES.)
      END
```

```
*DECK REFINE
      SUBROUTINE REFINE
*REFINE         REFINE THE GRID BY SUBDIVIDING THE INTERVALS -REFINE-

C     INPUT-
C      VMG1   = MAX VELOCITY DIFFERENCE BET GRID POINTS ALONG SL
C      VMG2   = MAX VELOCITY DIFFERENCE BET GRID POINTS ALONG OL
C      Z,R,PHI1,S1,S2,VM FIELD VALUES (PHI1 IS NOT PRESERVED)

       COMMON /ALLCOM/ MACHA,PSA,TSA,PTA,TTA, AXIA,RGA,GAMA,
      1                MACHC,PSC,TSC,PTC,TTC, AXIC,RGC,GAMC,
      2                DAXIT,SCALEA,TTE,CHOTST
        REAL           MACHA(1),MACHC
        LOGICAL        AXIA,AXIC
        LOGICAL        CHOTST
C      INDEX- M=MO,NM
       COMMON /CZ     / Z(300)
       COMMON /CR     / R(300)
       COMMON /CS2    / S2(300)
       COMMON /CS1    / S1(300)
       COMMON /CPHI1  / PHI1(300)
       COMMON /CM     / JMS(300)
       COMMON /CCURV  / CURV(300)
       COMMON /CB     / B(300)
       COMMON /CIDEX  / M,J,MU,MD,ISTAG
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
        DIMENSION      LIMITS(24)
        EQUIVALENCE    (LIMITS,LHO)
       COMMON /SLTAB / W(128),X2(128),SLCHN(128)
        INTEGER SLCHN
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
       COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
      1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
      1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
      3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
        LOGICAL        PRIM
        INTEGER TYPELB,TYPEUB
        DIMENSION      SCHOKE(1)
        EQUIVALENCE    (SCHOKE,DWDV)
        DIMENSION      IPRIM(1)
        EQUIVALENCE    (IPRIM,PRIM)

       COMMON /CBITS / BITS,BLANK
       COMMON /CCRX  / CRXSL,CRXOL,CRXSS,CRXE,CRXC,CRMACH
C      CRXSL  = NEW SL EXTENSION CRITERIA
C      CRXSS  = EXTENSION CRITERIA FOR NEW OL IN REGION WITH SOME SS-FLOW
C      CRXOL  = NEW OL EXTENSION CRITERIA
C      CRXE   = EXTENSION CRITERIA FOR NEW OL WHICH CROSSES SONIC LINE
C      CRXC   = EXTENSION CRITERIA FOR NEW OL WHICH CROSSES SHOCK WAVE
C      CRMACH= UPPER MACH NUMBER LIMIT FOR OL EXTENSION
       COMMON /CMAXIT/ MAXIT,MAJCTR,GREFIN,EDUM
```

```
      LOGICAL                    GREFIN
      COMMON /CPRINT/ PDUM1(3),PREFIN,PREFN2,SSONIC,PDUM(10)
      LOGICAL         PRTDB
      COMMON /CREFLE/ RLE1,RLE2,RLE3,HLE
      COMMON /CREFIN/SLS,SG21,VMG1,VMG2
     1,               NGR,NGZ, SGR(10),GR(10), SGZ(10),GZ(10)
      COMMON /CTABPR/ I1TAB
      COMMON /CTOLRL/ TOLRL(12),SG1REF,TOLINR
      COMMON /CVM    / VM(300)

      COMMON /ERASE2/ CR(128),DELS(128),DELVM(128),LSTA(128),MJ2(128),
     1                SGX(128),SGY(128),RAV(128),ZAV(128), IA(16),IB(16)
      INTEGER         EXT,FIELD,HINT,HLE,TE
      LOGICAL         DOWNB,EXTND1,EXTND2,HALVE,NEWSL,SSP,UPPER
      NAMELIST /NL2/ L1,L2,MA1,MB1,MAD1,MBD1, MA2,MB2,MAU2,MBU2
      NAMELIST /NL390/ S1A,S1B,S2A,S2B,S1A2,DOWNB
       NAMELIST /NL3/ X1NEW,DOWNB,L,L3,NPTS

      DATA FIELD/5HFIELD/

      GREFIN= .FALSE.
      QVMG1 = 1./VMG1
      QVMG2 = 1./VMG2
      IF(PDUM(7).NE.0.) CALL TABPRT(6HRFN-VM,VM,NM,NJ)

C     CHECK TO SEE IF PARTIAL OL SHOULD BE EXTENDED
C     CHECK TO SEE IF PARTIAL SL SHOULD BE EXTENDED
C      OMIT

C*** EXAMINE GRID INCREMENT BETWEEN ORTHOGONALS
  300 L1     = LO
      NAVG   = 0
      SG1AVG= 0.
      SG1MIN= 1.E6
      SGMX   = 0.
      SGMX2  = 0.

C     CHECK FOR ADJACENT STATIONS AND DETERMINE THE BASE STATION -
C       A BASE STATION IS THE OL UPSTREAM OF LE STAG PT,
C       DOWNSTREAM OF A TE, OR THE SHORTEST OF (PARTIAL) OL-S.
C       OTHERWISE THE BASE STATION CAN BE EITHER THE UPSTREAM OR DOWNSTRE
C      DOWNB = DOWNSTREAM BASE STATION
  305 L2     = L1+LNEXT(L1)
      IF(L2.GE.LESTA) GO TO 99
      MA1    = MLB(L1)
      M      = MA1
      CALL GETIX
      MAD1   = MD
      MB1    = MUB(L1)
      M      = MB1
      CALL GETIX
      MBD1   = MD
      MA2    = MLB(L2)
      M      = MA2
      CALL GETIX
      MAU2   = MU
      MB2    = MUB(L2)
      M      = MB2
```

```
      CALL GETIX
      MBU2  = MU
       PRTDB = .FALSE.
       IF(PREFN2.EQ.1. .OR. (PREFN2.GE.X1(L2).AND.X1(L1).GE.PDUM(8)
     1    .AND. (PREFN2+PDUM(8)).GT.0.)) PRTDB=.TRUE.
       IF(.NOT.PRTDB) GO TO 312
       CALL TABPRT(GHSTA-L1,X1(L1),1,1)
       WRITE (6,NL2)
  312  CONTINUE

C     ADJACENT STATION TEST
       IF((MA2.LE.MAD1 .AND. MAD1.LT.MB2)  .OR.
     1    (MA2.LT.MBD1 .AND. MBD1.LE.MB2)  .OR.
     2    (MA1.LE.MAU2 .AND. MAU2.LT.MB1)) GO TO 330

C         CHECK FOR TE FOLLOWED BY LE
          DATA TE/2HTE/
          IF(MAJCTR.GE.1) GO TO 550
          IF(TYPELB(L1).NE.TE) GO TO 322
          M      = MA1
          GO TO 324
  322     IF(TYPEUB(L1).NE.TE) GO TO 550
          M      = MB1
  324     CALL GETIX
          CALL STAX1(X1(L1),X2(J),X2(J),LXB,LXA)
C         LXB,LXA ARE STATIONS BELOW AND ABOVE THE TRAILING EDGE.
C         IF L2 IS A LEADING EDGE STATION FOLLOWING L1, THEN L1 MUST
C         BE THE SECOND OF THE TWO TE STATIONS.
          IF(L1.EQ.LXA .OR. L1.EQ.LXB) GO TO 325
            WRITE (6,NL2)
            CALL ERROR1
  325     IF(LXB.GT.L1 .OR. LXA.GT.L1) GO TO 550

C         INSERT AN ORTHOGONAL BETWEEN THE TRAILING EDGE AND
C           LEADING EDGE STATIONS.
C          DEFINE MJ2(I),CR(I),NI, DOWNB,L,L3
          I      = 0
          M      = MLB(LXB)
  326     I      = I+1
          MJ2(I)= M
          CR(I) = 2.
          M      = M+1
          IF(M.LE.MUB(LXB)) GO TO 326
          M      = MLB(LXA)
  327     I      = I+1
          MJ2(I)= M
          CR(I) = 2.
          M      = M+1
          IF(M.LE.MUB(LXA)) GO TO 327
          NI     = I
          DOWNB = .FALSE.
          L      = L1
          L3     = L2
       GO TO 440

C     NUMBER OF PRIMARY STATIONS
  330 NPRIM = 0
      IF( PRIM(L1).OR.PRIM(L2)) NPRIM=1
```

```
      IF( PRIM(L1).AND.PRIM(L2) ) NPRIM=2
      LBASE = L1
      IF(NPRIM-1) 340,350,360
C     NO PRIM STATIONS
  340 IF(MAU2.GT.MA1 .OR. MBU2.LT.MB1) GO TO 380
      GO TO 370
C     ONE PRIM STATION
  350 IF(PRIM(L1)) GO TO 380
      GO TO 370
C     BOTH L1 AND L2 ARE PRIM STATIONS
  360 IF((MB2-MA2).GT.(MB1-MA1)) GO TO 380

C     UPSTREAM BASE STATION
  370 DOWNB = .FALSE.
      MA    = MA1
      MB    = MB1
      L     = L1
      L3    = L2
      GO TO 390

C     DOWNSTREAM BASE STATION
  380 DOWNB= .TRUE.
      MA    = MA2
      MB    = MB2
      L     = L2
      L3    = L1


C     CHECK L.E. REFINEMENT CRITERIA
  390 IF(TYPELB(L3).NE.HLE .AND. TYPEUB(L3).NE.HLE) GO TO 395
C      NEW ORTHOGONAL IN FRONT OF L.E.
      IF(DOWNB) GO TO 394
      MX    = MBU2
      IF(TYPELB(L3).EQ.HLE) MX=MAU2-1
      S2B   = S2(MX)-S2(MX-1)
      S2A   = S2(MX+2)-S2(MX+1)
      M     = MX-1
      CALL GETIX
      S1B   = S1(MD)-S1(M)
      M     = MX+2
      CALL GETIX
      S1A   = S1(MD)-S1(M)
      IF(PREFN2.NE.0.) WRITE (6,NL390)
      IF((S1A.LE.RLE1*S2A .OR. S1B.LE.RLE1*S2B) .AND. MAJCTR.GE.1)
     1   GO TO 550
      GO TO 400
C      NEW ORTHOGONAL BEHIND L.E.
  394 M     = MB1-1
      IF(TYPELB(L3).EQ.HLE) M=MA1+1
      CALL GETIX
      S1A2  = S1(MD)-S1(M)
      DR    = R(M)-R(MU)
      DZ    = Z(M)-Z(MU)
      S1A   = SQRT(DZ*DZ+DR*DR)
      IF(PREFN2.NE.0.) WRITE (6,NL390)
      IF(S1A2.LE.RLE2*S1A .AND. MAJCTR.GE.1) GO TO 550
      GO TO 400
```

```
C     INHIBIT REFINEMENT AROUND A FIXED STAGNATION POINT
  395 M      = MLB(L3)
      CALL GETIX
      IF(ISTAG.NE.1) GO TO 399
      IF(DOWNB) GO TO 397
C      NEW OL IN FRONT OF STAG PT ON LOWER BDY
      S2A    = S2(MAU2+1)-S2(MAU2)
      M      = MAU2+1
  396 CALL GETIX
      S1A    = S1(MD)-S1(M)
      IF(S1A.LE.RLE1*S2A .AND. MAJCTR.GE.1) GO TO 550
      GO TO 400
C      NEW OL BEHIND STAG PT ON LOWER BDY
  397 M      = MA1+1
  398 CALL GETIX
      S1A2   = S1(MD)-S1(M)
      DR     = R(M)-R(MU)
      DZ     = Z(M)-Z(MU)
      S1A    = SQRT(DZ*DZ+DR*DR)
      IF(S1A2.LE.RLE2*S1A .AND. MAJCTR.GE.1) GO TO 550
      GO TO 400
C      NEW OL IN FRONT OF STAG PT ON UPPER BDY
  399 M      = MUB(L3)
      CALL GETIX
      IF(ISTAG.NE.1) GO TO 400
      IF(DOWNB) GO TO 3992
      S2A    = S2(MBU2)-S2(MBU2-1)
      M      = MBU2-1
      GO TO 396
C      NEW OL BEHIND STAG PT ON UPPER BDY
 3992 M      = MB1-1
      GO TO 398


C**  SWEEP ACROSS THE STREAMLINES TO CHECK FOR REQD GRID REFINEMENT
C      BETWEEN ORTHOGONALS L1 AND L2
  400 X1L3   = X1(L3)
      LX     = L1
      I      = 0
      M      = MA
      CRXL   = CRXOL
      SSP    = .FALSE.
  420 CALL GETIX
      MX     = MD
      IF(DOWNB) MX=MU
      IF(MX.EQ.0) GO TO 430
      CALL STAND(MX,LX,DUM)
      IF(X1(LX).NE.X1L3) GO TO 430
      I      = I+1
      DELS(I)= ABS(S1(MX)-S1(M))
C        CALC LARGEST, NEXT LARGEST DISTANCES BETWEEN ORTHOGONALS, SGMX,S
C         FOR DETERMINING NUMBER OF EXTRA SL-S
          IF(MAJCTR.GE.1) GO TO 425
          IF(DELS(I).LT.SGMX) GO TO 423
          SGMX2 = SGMX
          SGMX  = DELS(I)
          GO TO 425
```

250

```
  423     IF(DELS(I).GE.SGMX2) SGMX2=DELS(I)
C         MINIMUM DISTANCE BETWEEN ORTHOGONALS
  425     SG1MIN= AMIN1(SG1MIN,DELS(I))
C         AVERAGE DISTANCE BETWEEN ORTHOGONALS
          SG1AVG= SG1AVG+DELS(I)
          NAVG  = NAVG+1
        DELVM(I)=ABS(VM(MX)-VM(M))*QVMG1
        RAV(I)= .5*(R(MX)+R(M))
        ZAV(I)= .5*(Z(MX)+Z(M))
        MJ2(I)= M
C        CHECK FOR SUPERSONIC FLOW
        IF(B(M).LT.0. .OR. B(MX).LT.0.) SSP=.TRUE.
C        COMPUTE ORIENTATION OF SONIC LINE
        IF(PRTDB) WRITE(6,1426) M,MX,B(M),B(MX)
 1426 FORMAT(24H REFINE-M,MX,B(M),B(MX)-2I6,2F10.3)
        IF(M.EQ.MA) GO TO 430
        BM    = B(M)
        BMM1  = B(M-1)
        IF(M.NE.(MA+1) .OR. B(MX-1)*BMM1.GE.0.) GO TO 4258
        F     = 0.
        BMM1  = B(MX-1)
        GO TO 4260
 4258 IF(M.NE.MB .OR. B(MX)*BM.GE.0.) GO TO 4259
        F     = 1.
        BM    = B(MX)
 4259 IF(BM*BMM1.GT.0.) GO TO 430
        F     = BMM1/(BMM1-BM)
 4260 FX    = 0.
 4261 IF(B(MX)*BM.GE.0.) GO TO 4262
        FX    = FX+1.
        IF(MX.GE.MUB(L3)) GO TO 4265
        MX    = MX+1
        GO TO 4261
 4262 IF(B(MX-1)*BMM1.GE.0.) GO TO 4264
        FX    = FX-1.
        IF((MX-1).LE.MLB(L3)) GO TO 4265
        MX    = MX-1
        GO TO 4262
 4264 FX    = B(MX-1)/(B(MX-1)-B(MX)) + FX
 4265 FX    = FX-F
        IF(PRTDB) WRITE(6,1427) MX,F,FX
 1427 FORMAT(9H MX,F,FX-I6,2F10.3)
        IF(DOWNB) FX=-FX
C       FX    = SLOPE OF SONIC LINE IN TERMS OF GRID INTERVALS
C         CHECK FOR EXPANSION OR COMPRESSION
        IF(BM*FX.LT.0.) GO TO 4266
C          EXPANSION
        CRXL   = AMIN1(CRXL,CRXE)
        FX    = ABS(FX)*SLS
        IF(FX.GE..75) DELS(I)=AMAX1(DELS(I),(S2(M)-S2(M-1))/SG21)
        DELVM(I)=AMAX1(DELVM(I),10.*(FX-1.))
C          COMPRESSION
 4266 CRXL   = AMIN1(CRXL,CRXC)
  430 M     = M+1
        IF(M.LE.MB) GO TO 420
        IF(CRXSS.LE.CRXL .AND. SSP) CRXL=CRXSS
        NI    = I
        CALL LFIT1(GR,SGR,NGR, RAV,SGY,NI)
```

```
      CALL LFIT1(GZ,SGZ,NGZ, ZAV,SGX,NI)
      HALVE = .FALSE.
      DO 432 I=1,NI
      RS    = DELS(I)/AMAX1(SGX(I),SGY(I))
      CR(I) = RS + DELVM(I)*RS**.2
  432 IF(CR(I).GT.1.) HALVE=.TRUE.
       IF(.NOT.PRTDB) GO TO 435
       CALL TABPRT(3HMJ2,MJ2,NI,10)
       CALL TABPRT(3HRAV,RAV,NI,10)
       CALL TABPRT(3HZAV,ZAV,NI,10)
       CALL TABPRT(3HSGX,SGX,NI,10)
       CALL TABPRT(3HSGY,SGY,NI,10)
       CALL TABPRT(4HDELS,DELS,NI,10)
       CALL TABPRT(5HDELVM,DELVM,NI,10)
       CALL TABPRT(2HCR,CR,NI,10)
  435  CONTINUE

C     PREVENT TOO RAPID CHANGE IN OL SPACING
       IF(HALVE) GO TO 440
       X1D12 = .5*(X1(L2)-X1(L1))
       IF(PRIM(L1)) GO TO 436
       IF((X1(L1)-X1(L1M)).LT.X1D12) HALVE=.TRUE.
       GO TO 437
  436 L1M    = L1
  437 IF(PRIM(L2)) GO TO 438
       L2P    = L2+LNEXT(L2)
       IF((X1(L2P)-X1(L2)).LT.X1D12) HALVE=.TRUE.
       GO TO 439
  438 L2P    = L2
  439 IF(.NOT.HALVE) GO TO 550
       IF(TYPELB(L1).EQ.FIELD .OR. TYPELB(L1M).EQ.FIELD .OR.
      *    TYPELB(L2).EQ.FIELD .OR. TYPELB(L2P).EQ.FIELD) GO TO 4391
       CR(1) = 1.
       GO TO 440
 4391 CR(NI)=1.

C**   ADD A NEW ORTHOGONAL LINE BETWEEN L1 AND L2, FIRST CHECK MEMORY
  440 X1NEW = .5*(X1(L1)+X1(L2))
       EXTND1= .TRUE.
       EXTND2= .TRUE.
       IF(TYPELB(L).EQ.FIELD) EXTND1=.FALSE.
       IF(TYPEUB(L).EQ.FIELD) EXTND2=.FALSE.
       IRET   = 0
       IF((LESTA+20).LE.MAXLE) GO TO 800
       WRITE (6,1440) X1NEW
 1440 FORMAT(1X72H***  STATION TABLE STORAGE LIMIT DOES NOT ALLOW A NEW
      *ORTHOGONAL AT X11=F7.3,1H./6X61HGRID REFINEMENT BY INSERTING ORTHO
      *GONALS IS BEING TERMINATED.)
       GO TO 99
  450 IF(NL.EQ.1) GO TO 455
       WRITE (6,1450) NL,X1NEW
 1450 FORMAT(/3X,I2,1X17HOL-S REQUESTED ATF8.3,)
       IB(1) = IB(NL)
       NL    = 1

C**   ADJUST FIELD ARRAYS FOR THE NEW OL
  455 NPTS   = IB(1)-IA(1)+1
       GREFIN= .TRUE.
```

```
        IF(PRTDB) WRITE(6,NL3)
       CALL ADDFPT(MA2,NPTS,999999)

C     CORRECT THE POINTERS IN THE JMS-TABLE
       MNEW   = MA2
       MA     = MNEW
       I      = IA(1)
  460 IF(DOWNB) GO TO 470

C      (UPSTREAM BASE STATION)
C      UPSTREM POINT
       M      = MJ2(I)
       CALL GETIX
       MDSAV = MD
       MD     = MNEW
       CALL SAVIX
C      NEW POINT
       MU     = M
       M      = MNEW
       MD     = MDSAV
       ISTAG = 0
       CALL SAVIX
C      DOWNSTREAM POINT
       M      = MD
       CALL GETIX
       MU     = MNEW
       CALL SAVIX
       GO TO 490

C      (DOWNSTREAM BASE STATION)
C      DOWNSTREAM POINT
  470 M      = MJ2(I)+NPTS
       CALL GETIX
       MUSAV = MU
       MU     = MNEW
       CALL SAVIX
C      NEW POINT
       MD     = M
       M      = MNEW
       MU     = MUSAV
       ISTAG = 0
       CALL SAVIX
C      UPSTREAM POINT
       M      = MU
       CALL GETIX
       MD     = MNEW
       CALL SAVIX

  490 I      = I+1
       MNEW   = MNEW+1
       IF(IB(1)-I) 495,460,460
  495 MB     = MNEW-1
        IF(PRTDB) CALL JMSPRT

C**  MODIFY STATION-TABLE
  500 CALL INSTA(L2,L,L3,DOWNB, MA,MB)

C     INCREMENT TO THE NEXT ORTHOGONAL INTERVAL
```

```
  550 LIM    = L1
      L1     = L2
      GO TO 305

C     AVERAGE DIST BET ORTHOGS
   99 SG1AVG= SG1AVG/FLOAT(NAVG)
      SG1RLF= .5*(SG1MIN+SG1AVG)

C*** EXAMINE GRID INCREMENT ABOVE STREAMLINE J2, (J2=1,NJ)
      J2     = 1
       IF(PREFN2.EQ.0.) GO TO 100
       I1TAB = LO
       CALL TABPRT(6H560STA,X1,LESTA,5)
  100 J2NEXT= J2+1
      IF(W(J2+1).EQ.0.) GO TO 200
C        NEXTRA= NO OF EXTRA SL-S NEAR THE BODY FOR CHN=EXT,INT
         NEXTRA= 0
         DATA EXT/3HEXT/, HINT/3HINT/
         IF(MAJCTR.GT.0 .OR. (SLCHN(J2).NE.EXT .AND. SLCHN(J2).NE.HINT))
     1         GO TO 104
         M      = MBEGIN(J2)
         DSOL   = SGMX2/2.
         RROL   = (R(M+1)-R(M))/DSOL
         IF(AXIA) RROL=(R(M+1)*R(M+1)-R(M)*R(M))/(DSOL*(R(M)+DSOL))
         RR     = 0.
         IF(R(M).LE..1) GO TO 101
C        THE FIRST SL IS TO BE PLACED ABOUT ONE BODY RADIUS AWAY
         RRATIO= R(M+1)/R(M)
         RR     = RRATIO-1.
         IF(AXIA) RR=(RRATIO*RRATIO-1.)/3.
  101    RR     = AMAX1(RR,RROL)
C        NEXTRA= MAXO(1,MINO(INT(ALOG(RR)/ALOG(2.))-1,8))
         NEXTRA= MAXO(1,INT(ALOG(RR)/ALOG(2.)))
  104 M      = MBEGIN(J2)
C     M      = THE FIRST POINT ON THE STREAMLINE
      EXTND1= .TRUE.
      EXTND2= .TRUE.
      L      = 0
      WMIN   = 1.E6
      I      = 1
  110 CALL GETIX
      MNEXT = MD
      CALL STAND(M,L,UPPER)
C      BYPASS UPPER BOUNDARY OF PARTIAL OL
      IF(UPPER) GO TO 120
C      CHECK L.E. REFINEMENT CRITERIA
       IF(ISTAG.NE.1) GO TO 114
       S2A    = S2(MU+1)-S2(MU)
       DZ     = Z(M+1)-Z(MU+1)
       DR     = R(M+1)-R(MU+1)
       S1A    = SQRT(DZ*DZ+DR*DR)
       DZ     = Z(MD+1)-Z(M+1)
       DR     = R(MD+1)-R(M+1)
       S1A2   = SQRT(DZ*DZ+DR*DR)
       IF((S2A.LT.RLE3*S1A .OR. S2A.LT.RLE3*S1A2) .AND. MAJCTR.GE.1)
     1 GO TO 200
  114 LSTA(I)=L
      MJ2(I)= M
```

254

```
      DELS(I)=S2(M+1)-S2(M)
C     (NOTE-S2 IS NOT UPDATED IF THIS IS FOR AN EXTRA SL)
      DELVM(I)=ABS(VM(M+1)-VM(M))*QVMG2
      ZAV(I)= .5*(Z(M+1)+Z(M))
      RAV(I)= .5*(R(M+1)+R(M))
      M     = M+1
      CALL GETIX
      IF(I.EQ.1 .AND. MU.NE.0) EXTND1=.FALSE.
      IF(MNEXT.EQ.0 .AND. MD.NE.0) EXTND2=.FALSE.
C      CHECK L.E. REFINEMENT CRITERIA
       IF(ISTAG.NE.1) GO TO 117
       S2B   = S2(MU)-S2(MU-1)
       DZ    = Z(MU-1)-Z(M-1)
       DR    = R(MU-1)-R(M-1)
       S1B   = SQRT(DZ*DZ+DR*DR)
       DZ    = Z(MD-1)-Z(M-1)
       DR    = R(MD-1)-R(M-1)
       S1B2  = SQRT(DZ*DZ+DR*DR)
       IF((S2B.LT.RLE3*S1B .OR. S2B.LT.RLE3*S1B2) .AND. MAJCTR.GE.1)
     1 GO TO 200
  117 IF(W(J).GE.WMIN) GO TO 119
      WMIN  = W(J)
      X2MIN = X2(J)
  119 I     = I+1
  120 M     = MNEXT
      IF(M.NE.0) GO TO 110
      NI    = I-1
      CALL LFIT1(GR,SGR,NGR, RAV,SGY,NI)
      CALL LFIT1(GZ,SGZ,NGZ, ZAV,SGX,NI)
C     CR(I)=1 IS THE RADIUS OF PERMISSIBLE GRID SIZE
      HALVE = .FALSE.
      DO 132 I=1,NI
      RS    = ABS(DELS(I))/(AMAX1(SGX(I),SGY(I))*SG21)
      CR(I) = RS + DELVM(I)*RS**.2
      IF(CR(I).GT.1.) HALVE=.TRUE.
  132 CONTINUE

C*** IF HALVE=T ADD NEW SL FOR STATIONS FOR WHICH CR.GT.5
       PRTDB = .FALSE.
       IF(PREFIN.EQ.1. .OR. (PREFIN.GT.X2(J2).AND.X2(J2).GE.PDUM(8)))
     1   PRTDB=.TRUE.
       IF(.NOT.PRTDB) GO TO 141
      CALL TABPRT(2HCR,CR,NI,10,0)
      CALL TABPRT(3HMJ2,MJ2,NI,10,0)
      CALL TABPRT(4HLSTA,LSTA,NI,10,0)
  141 CONTINUE
      IF(.NOT.HALVE) GO TO 200
      IRET  = -1
      CRXL  = CRXSL
      GO TO 800
  145 WNEW  = .5*(W(J2)+WMIN)
      XI2   = .5*(X2(J2)+X2MIN)
       IF(PRTDB) CALL TABPRT(4HWNEW,WNEW,1,1)

C     BEGIN LOOP FOR INSERTING THE (PARTIAL) STREAMLINE, LI=1,NL
      LI    = 1
      NPTADD= 0
  150 I1    = IA(LI)
```

255

```
      I2     = IB(LI)
      IF(I1.EQ.0) GO TO 195

C     DETERMINE J1, INDEX OF NEW SL
      J      = J2
  160 IF(W(J).GT.WNEW) GO TO 170
      J      = J+1
      IF(J.GT.NJ) CALL ERROR1
      GO TO 160
  170 J1     = J

C     ADJUST FIELD ARRAYS AND  SL TABLES
      NEWSL  = .TRUE.
      I      = I1
      MU1    = 0
      IF(NJ.LT.MAXNJ) GO TO 180
      WRITE (6,1175) XI2
      RETURN
  180 L      = LSTA(I)
      M1     = MJ2(I)+NPTADD+1
      MD1    = 0
      CALL ADPTSL(M1,MU1,MD1,J1,NEWSL)
      NPTADD= NPTADD+1
      M      = M1+1
      CALL GETIX
      JP     = J
      M      = M1-1
      CALL GETIX
      JM     = J
      M      = M1
      J      = J1
      W(J)   = WNEW
      X2(J)  = XI2
      F      = (W(JP)-WNEW)/(W(JP)-W(JM))
      ONEMF  = 1.-F
      M      = M1
      B(M)   = B(M-1)*F + B(M+1)*ONEMF
      R(M)   = R(M-1)*F + R(M+1)*ONEMF
      IF(AXIA) R(M)=SQRT(R(M-1)*R(M-1)*F + R(M+1)*R(M+1)*ONEMF)
      S1(M)  = S1(M-1)*F + S1(M+1)*ONEMF
      VM(M)  = VM(M-1)*F + VM(M+1)*ONEMF
      Z(M)   = Z(M-1)*F + Z(M+1)*ONEMF
C       SET ISTAG=3 FOR PTS ADJACENT TO L.E. AND BOUNDARY CORNER PTS.
        IF(.NOT.PRIM(L)) GO TO 185
        M      = M1-1
        CALL GETIX
        ISTAGM= ISTAG
        M      = M1+1
        CALL GETIX
        IF(ISTAGM.EQ.1) GO TO 181
        IF(ISTAG.NE.1) GO TO 185
C       (ISTAGP=1)
        ISTAGM= 0
        GO TO 182
C       (ISTAGM=1)
  181   ISTAG = 0
        CALL SAVIX
  182   M      = M1
```

256

```
        CALL GETIX
        ISTAG = 3
        CALL SAVIX
        M     = M1-1
        CALL GETIX
        ISTAG = ISTAGM
        CALL SAVIX

C     UPDATE THE STATION-TABLE POINTERS TO THE FIELD-TABLE
  185 CALL STTOFI(L,1)
      GREFIN= .TRUE.

C     INDEX TO NEXT PT ON SL
      NEWSL = .FALSE.
  190 I     = I+1
      MU1   = M1
      IF(I2-I) 194,180,180

C     INDEX TO NEXT PARTIAL SL
  194 J2NEXT= J2NEXT+1
  195 LI    = LI+1
       IF(.NOT.PRTDB) GO TO 197
      IITAB = LO
      CALL TABPRT(6H195STA,X1,LESTA,5)
      CALL JMSPRT
  197 CONTINUE
      IF(NL-LI) 200,150,150

C     LOOP TO PUT IN ADDITIONAL SL-S FOR EXTERNAL CHANNELS
  200 IF(NEXTRA.EQ.0) GO TO 210
      NEXTRA= NEXTRA-1
      GO TO 104

C     INCREMENT THE STREAMLINE COUNTER J2
  210 J2    = J2NEXT
      IF(J2.LT.NJ) GO TO 100
      RETURN


C*** EVALUATION OF NEW LINE POSITIONS
C     OUTPUT-
C      NL NEW LINES ARE TO BE IN THE REGIONS IA(LI) TO IB(LI), LI=1,NL
C      FOR IA(LI).NE.0.

C     SEARCH FOR CR.GT.1. POINT
  800 NL    = 0
      I     = 1
  805 IF(CR(I).GE.1.) GO TO 810
      I     = I+1
      IF(I.LE.NI) GO TO 805
      GO TO 840

C     FIND  IA,IB  SO THAT  CR.GE..375 IS WITHIN  IA,IB
  810 NL    = MINO(NL+1,10)
      ISAVE = I
  815 IA(NL)= I
      I     = I-1
      IF(I.GE.1 .AND. (I.GE.(ISAVE-3).OR.CR(I).GE.CRXL)) GO TO 815
```

```
      I      = ISAVE
  820 IB(NL) = I
      I      = I+1
      IF(I.GT.NI) GO TO 840
      IF(CR(I).GE.1.) ISAVE=I
      IF(I.LT.(ISAVE+3) .OR. CR(I).GE.CRXL) GO TO 820

C     REPEAT THE ABOVE FOR THE NEXT PARTIAL LINE
      IF(I.LT.NI) GO TO 805

C     ADD ONLY ONE LINE IF NL.EQ.10
  840 IF(NL.NE.10) GO TO 850
      NL     = 1
      IB(1) = IB(10)

C     ELIMINATE THE SHORT GAPS BETWEEN LINES
  850 IF(NL.LE.1) GO TO 860
      LILAST= 1
      DO 855 LI=2,NL
      IF((IA(LI)-IB(LI-1)).GT.7) GO TO 854
      IB(LI-1)=IB(LI)
      IA(LI)= 0
      GO TO 855
  854 LILAST= LI
  855 CONTINUE
      NL     = LILAST
  860 IF(IA(1).LE.2 .AND. EXTND1) IA(1)=1
      IF((NI-IB(NL)).LE.2 .AND. EXTND2) IB(NL)=NI

C     EXTEND EACH LINE TO A MINIMUM OF FIVE POINTS
      NPTS   = 0
      DO 870 LI=1,NL
      IF(IA(LI).EQ.0) GO TO 870
  865 IDEF   = MAXO((5-(IB(LI)-IA(LI)))/2, 0)
      IA(LI)= MAXO(IA(LI)-IDEF,1)
      IB(LI)= MINO(IB(LI)+IDEF,NI)
      NPTS   = NPTS + IB(LI)-IA(LI)+1
      IF(NPTS.LT.5 .AND. NPTS.LT.NI) GO TO 865
  870 CONTINUE
       IF(PRTDB) WRITE (6,1880) (IA(LI),IB(LI),LI=1,NL)
 1880 FORMAT(7H IA,IB=,10(4X,2I4))
      IF((NM+NPTS).LE.MAXNM) GO TO 890
      WRITE (6,1881) NM,MAXNM
      RETURN

C     RETURN
  890 IF(IRET) 145,450,450

 1175 FORMAT(/2X36H*** STREAMLINE LIMIT REACHED.  (XI2=F6.3,1H))
 1881 FORMAT(/2X69H*** FIELD POINT STORAGE LIMIT PREVENTS FURTHER GRID R
     *EFINEMENT.  (NM=I4,8H, MAXNM=I4,1H))
      END
```

256

```
*DECK REFBLK
      BLOCK DATA REFBLK
*REFBLK          BLOCK DATA FOR REFINE                          -REFBLK-
      COMMON /CREFLE/ RLE1,RLE2,RLE3,HLE
      DATA RLE1,RLE2,RLE3/.65,1.3,1.3/, HLE/2HLE/
      END
```

```
*DECK SLC
      SUBROUTINE SLC
*SLC---          STREAMLINE CURVATURE ETC                          -SLC-

C****CALCULATE ANGLE, CURVATURE AND ARC LENGTH ALONG STREAMLINES


C     INPUT-
C      B      = SUBSONIC SUPERSONIC INDICATOR, NEGATIVE FOR SUPERSONIC VEL
C      Z,R    = STREAMLINE COORDINATES

C     OUTPUT-
C      PHI1   = ANGLE IN RADIANS
C      CURV   = CURVATURE
C      S1     = ARC LENGTH

       COMMON /CBEAM2/ DR,DZ,YPA,YPB,F,G, DX,YQDX,ZM,RM,ANGM,CURVM,S1M,
      1                RZONLY, ANGCHD,SINTVL, YPASQ,YPAB,YPBSQ
        LOGICAL        RZONLY
       COMMON /CSS    / SSFML,SSEF,SSEANG,SSDF,SSFEND,SSFND1
      1                ,SSDLE,A4FACT,BRLX,CURRLX
        INTEGER        SSFML
        LOGICAL              SSEF,       SSDF,   SSDLE
C      SSFML = SUPERSONIC CURVATURE FORMULA NUMBER
C      SSEF  = SUPERSONIC ENTERING FLOW, T OR F
C      SSEANG= ENTERING FLOW ANGLE (DEGREES) FOR SSEF=T
C      SSDF  = SUPERSONIC DISCHARGE FLOW, T OR F
C      SSFEND= SUPERSONIC BEAM DOWNSTREAM END CONDITION, =0,1 FOR PARABOL
C      SSFND1= SUPERSONIC BEAM UPSTREAM END CONDITION, =0,1, FOR PARABOLA
C      SSDLE = SS FLOW BELOW AND AFT OF LE PT, T OR F
C      A4FACT= CENTRAL POINT INFLUENCE COEFFICIENT FACTOR
C      BRLX  = B-RELAXATION FACTOR
C      CURRLX= CURVATURE RELAXATION FACTOR
C      INDEX- M=MO,NM
       COMMON /CZ     / Z(300)
       COMMON /CR     / R(300)
       COMMON /CS2    / S2(300)
       COMMON /CS1    / S1(300)
       COMMON /CPHI1  / PHI1(300)
       COMMON /CM     / JMS(300)
       COMMON /CCURV  / CURV(300)
       COMMON /CB     / B(300)
       COMMON /CIDEX  / M,J,MU,MD,ISTAG
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
        DIMENSION      LIMITS(24)
        EQUIVALENCE    (LIMITS,LHO)
        EQUIVALENCE    (LDUM(1),LSO), (LDUM(2),LSE)
       COMMON /SLTAB / W(128),X2(128),SLCHN(128)
        INTEGER        SLCHN
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL   = SHARP CORNER INDICATOR (BLDTBS)
C      MCL   = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
       COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
```

```
     1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
     1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
     3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
       LOGICAL        PRIM
       INTEGER TYPELB,TYPEUB
       DIMENSION      SCHOKE(1)
       EQUIVALENCE    (SCHOKE,DWDV)

       DIMENSION      BDT(1),LBNEXT(1),LBZ1(1),CHNAME(1),UP(1),
     1                LEDEX(1),ZBT(1),RBT(1),ANGBT(42)
       INTEGER        BDT,CHNAME,BDNAME
        LOGICAL       UP
        DIMENSION     BDNAME(1),LBA(1),LBB(1)
       EQUIVALENCE    (BDNAME,ZBT), (LBA,RBT), (LBB,ANGBT)
       EQUIVALENCE    (X1,BDT), (LNEXT,LBNEXT), (MLB,LBZ1), (MUB,CHNAME)
       EQUIVALENCE    (PRIM,UP), (TYPELB,LEDEX), (NAMELB,ZBT)
       EQUIVALENCE    (ILB,RBT), (FLB,ANGBT)

      COMMON /CBEAM / DBEAM(3),IORDER
      COMMON /BENDIN/ NBCIN(2),ACF(2)
      COMMON /CBEND / NBCB(2),ANGE(2),CURVE(2),FB(2)
      COMMON /CBITS / BITS,BLANK
      COMMON /CBDYPT/ ANGD,CURVD
      COMMON /CFB   / L,MA,MB,J2,IA,IB,I
      COMMON /CINNER/ INRCTR
      COMMON /CMAXIT/ MAXIT,MAJCTR,GREFIN,EDUM
      COMMON /CPI   / PI,TWOPI,PIQ2,PIQ4,TODEG,TORAD
      COMMON /CPRINT/ PDUMX(6),PDUM(20)
      COMMON /CPTMOV/ VELPOT,ICOB,NODENS,FBASTG
      COMMON /CQIREM/ YTOL,YO,DYDX,CTRMAX
      COMMON /CSLC  / BRANCH(4)
      COMMON /CTABPR/ I1TAB
      COMMON /ERASE / A(3),BA(1),BB(1),DA(1),ACHD(1),CHD(793)
      COMMON /ERASE2/ RB(128),ZB(128),ANG(128),CURVB(128),S1B(128),
     1                BI(128),J2DONE(128),MSV(128),CURSS(6),QV(8)
      LOGICAL         ALLJ2,ANYJ2,J2PREV,PARSLA,UPPER

      DATA LE/2HLE/



C     FIRST PASS ACROSS STREAMLINES, SKIP THOSE SL-S WHICH TERMINATE WITH
C      IN THE FIELD IF J2PREV=T.  AT END OF PASS  ALLJ2=T IF ALL STREAMLI
C      HAVE BEEN FITTED AND ANYJ2=T IF ONE OR MORE SL-S HAVE BEEN FITTED.
C      J2PREV=F IF ON THE PREVIOUS PASS NO SL-S WERE FITTED BECAUSE END
C      CONDITION INTERPOLATION REQUIREMENTS COULD NOT BE SATISFIED.
      ANYJ2 = .TRUE.
      IF(PDUM(1).GT.0.) WRITE(6,1159)
      CALL SETM(1,0, J2DONE,NJ)
      RZONLY= .FALSE.

C     BEGIN LOOP THROUGH FIRST TO LAST STREAMLINE, J2=1,NJ
C      CALL MBEGIN TO OBTAIN FIELD INDEX OF FIRST PT ON SL
  100 J2PREV= ANYJ2
      ANYJ2 = .FALSE.
      ALLJ2 = .TRUE.
      J2    = 1
  101 IF(J2DONE(J2).EQ.1)  GO TO 187
```

```
      M      = MBEGIN(J2)
      IF(PDUM(1).GT.0.) WRITE (6,1160) J2

C     CHECK FOR ORTHOGONAL TERMINATING ON BOW SHOCK (DUMMY STREAMLINE )
      CALL GETIX
      IF( MU.EQ.0 .AND. MD.EQ.0 ) GO TO 180

C     BUILD ZB,RB,ANG ARRAYS FOR THE STREAMLINE SEGMENT
C      ISTAG=3 IS A BOUNDARY OF A PARTIAL ORTHOGONAL, SUCH POINTS
C      ARE TO BE BYPASSED AND THEN FILLED IN BY INTERPOLATION
  115 I      = 1
      S1B(1)= 0.
  120 IA     = I
      MA     = M
  121 CALL GETIX
      IF(ISTAG.EQ.3) GO TO 128
      RB(I) = R(M)
      ZB(I) = Z(M)
      ANG(I)= PHI1(M)
      BI(I) = B(M)
      MSV(I)= M
      IF(ISTAG.EQ.1 .OR. ISTAG.EQ.2) GO TO 130
  124 IF(MD) 126,130,126
  126 I      = I+1
      IB     = I
  128 M      = MD
      MB     = M
      GO TO 121

C     SET END CONDITIONS
  130 NBCB(1)=0
      NBCB(2)=0
      FB(1) = 0.
      FB(2) = 0.
      L      = 0
      MDSV   = MD
      ISTAGB= ISTAG
C      PARSLA= PARTIAL STREAMLINE AT END A, T OR F
      PARSLA= .FALSE.
      IEND   = 1
      MX     = MA
      IF(IA.EQ.1) GO TO 2140
 2135 IEND   = 2
      MX     = MB
      IF(MDSV.NE.0) GO TO 135
C       USE AVG CURVATURE B.C. FOR PARTIAL SL-S
 2140 CALL STAND(MX,L,UPPER)
      IF(MX.EQ.MLB(L) .OR. UPPER .OR.
     1   L.EQ.LO .OR. (L+LNEXT(L)).GE.LESTA)  GO TO 2180
      M    = MLB(L)
      CALL GETIX
      IF(MU.EQ.0 .OR. MD.EQ.0) GO TO 2180
C           PARTIAL SL, SEARCH FOR NON-TERMINATING ADJACENT SL
            SUM    = 0.
            CURVX  = 0.
            M      = MX
            MCHNG  = -1
 2150       M      = M+MCHNG
```

```
            CALL GETIX
            IF(MU.EQ.0 .OR. MD.EQ.0) GO TO 2150
            IF(J2DONE(J).EQ.0 .AND. J2PREV) GO TO 186
            IF(INRCTR.NE.0) GO TO 2155
            IF(J2DONE(J).EQ.0) GO TO 2150
 2155       IF(M.LT.MLB(L) .OR. M.GT.MUB(L)) GO TO 2157
            SUM   = SUM+1.
            CURVX = CURVX+.5*CURV(M)
 2157       IF(MCHNG.EQ.1) GO TO 2160
            M     = MX
            MCHNG = 1
            GO TO 2150
 2160       CURVX = CURVX/SUM
            NBCB(IEND)=2
            CURVE(IEND)=CURVX
            IF(IEND.EQ.1) PARSLA=.TRUE.
            GO TO 2190
 2180 NBCB(IEND)=NBCIN(IEND)
      ANGE(IEND)=ACF(IEND)
      CURVE(IEND)=ACF(IEND)
      FB(IEND)=ACF(IEND)
 2190 IF(IEND.EQ.1) GO TO 2135

C     DEFINE ANG(1)  TO OBTAIN CORRECT ANGLE BRANCH
  135 IF(IA.NE.1) GO TO 136
      ANG(1)= BRANCH
      IF(BRANCH.NE.999.) GO TO 136
      L     = 0
      M     = MSV(1)
      CALL STANO(M,L,UPPER)
      IF(M.NE.MLB(L)) GO TO 1352
C      FIRST STREAMLINE
      LB    = LBF(NAMELB(L))
      LB    = LB+LBZ1(LB)
      ANG(1)= ANGBT(LB)
      GO TO 136
C      NOT FIRST STREAMLINE
 1352 M     = M-1
      IF(M.LT.MLB(L)) CALL ERROR1
      CALL GETIX
      IF(J2DONE(J).EQ.0) GO TO 1352
      ANG(1)= PHI1(M)
      IF(PDUM(19).EQ.1.) WRITE (6,1353)J,M,ANG(1)
 1353 FORMAT (8H J,M,ANG,2I6,F10.6)
  136 IF(ISTAGB.NE.1) GO TO 155

C     THE STREAMLINE IS TERMINATED BY A STAGNATION POINT.
C      PROCEED TO EXTRAPOLATE FOR ITS POSITION IF STAG=1
C      AND BOUNDARY TYPE=LE.

C     FIND THE STAGNATION POINT STATION
      L     = 0
      CALL STANO(MB,L,UPPER)

C     CHECK FOR LEADING EDGE POINT
      CURVD = 0.
      IF(UPPER) GO TO 138
      IF(TYPELB(L).NE.LE) GO TO 155
```

```
      GO TO 140
  138 IF(TYPEUB(L).NE.LE) GO TO 155

C     BEGIN ITERATION FOR STAGNATION POSITION
  140 QV(1) = 0.
      SMOVE = 0.
      M     = MB
      IF(ABS(PDUM(5)).LT.5.) FB(2)=1.
  145 IF(UPPER) GO TO 147
      NAMES = NAMELB(L)
      IBS   = ILB(L)
      FS    = FLB(L)
      SIS   = SILB(L)
      GO TO 148
  147 NAMES = NAMEUB(L)
      IBS   = IUB(L)
      FS    = FUB(L)
      SIS   = SIUB(L)
  148 CALL BDYPTM(NAMES,IBS,ZB(I),RB(I),FS,SIS,SMOVE,GETASK)
      IF(GETASK.NE.0.) CALL ERROR1
      Z(M)  = ZB(I)
      R(M)  = RB(I)
      CALL BFACS(ZB,RB,ANG,CURVB,SIB, IA,IB)
C      (LOGIC FOR LEADING STAGNATION POINT ONLY)
      ERRANG= ANG(I)-(ANGD-PIQ2)

      IF(PDUM(1).LE.0.) GO TO 150
      WRITE (6,1149) QV(1),SMOVE,ERRANG,ZB(I),RB(I),ANGD,CURVD
 1149 FORMAT(14H STAG PT - QV=F5.0,2X,6HSMOVE=F10.5,2X,7HERRANG=F10.6,2X
     *,3HZD=F10.5,2X,3HRD=F10.5,2X,5HANGD=F10.3,2X,6HCURVD=F10.6)
      GO TO 1501
  150 IF(CURVD.GE.0.) GO TO 1501
      WRITE (6,1150) ZB(I),RB(I),ANGD,CURVD
 1150 FORMAT(/35H ***  NEGATIVE L.E. CURVATURE-    Z=F10.5,3X,2HR=F10.5,3
     *X,4HANG=F10.3,3X,5HCURV=F12.6)

 1501 IF(QV(1).NE.0.) GO TO 151
      YO    = 0.
      YTOL  = 1.E-5
      DYDX  = ABS(CURVD) + 1./(SIB(I)-SIB(I-1))
      XJP   = -ABS(ERRANG)/DYDX
      DYDX  = 0.
  151 CALL QIREM(SMOVE,ERRANG, XJP,QV)
      IF(QV(1).NE.0.) GO TO 145
      IF(UPPER) GO TO 152
      ILB(L)= IBS
      FLB(L)= FS
      SILB(L)=SIS
      GO TO 156
  152 IUB(L)= IBS
      FUB(L)= FS
      SIUB(L)=SIS
      GO TO 156

C     USE (SUBSONIC) BEAM FORMULA TO CALC ANG,CURVATURE,SI
C     SET IORDER=1 TO CHECK FOR POINT ORDERING
  155 NORDER= 1
 1552 IORDER= 1
```

```
      ANG(1)= 0.
      CALL BFACS(ZB,RB,ANG,CURVB,S1B, IA,IB)
      IF(IORDER.EQ.0) GO TO 1556
      I     = IORDER-1
      WRITE(6,1155) ZB(I),RB(I),ZB(I+1),RB(I+1),J2,I,IORDER
      IF(NORDER.GE.5) CALL ERROR1
      SAV   = ZB(I)
      ZB(I) = ZB(I+1)
      ZB(I+1)=SAV
      SAV   = RB(I)
      RB(I) = RB(I+1)
      RB(I+1)=SAV
      NORDER= NORDER+1
      GO TO 1552
 1556 IF(SSFML.EQ.(-1))CALL BF3(ZB,RB,ANG,CURVB,IA,IB)
  156 IF(SSEF .AND. .NOT.PARSLA) ANG(1)=SSEANG*TORAD

C     RELOCATE ANSWERS INTO FIELD STORAGE
  160 M     = MA
      I     = IA-1
      L     = 0
  161 CALL GETIX
      IF(ISTAG.EQ.3) GO TO 166
      I     = I+1
C           SUPERSONIC POINT CURVATURE
            IF(B(M).GE.0. .OR. I.EQ.1) GO TO 163
           IX    = 1
            I1SS  = I-1-IABS(SSFML)
            NBCB(1)=0
            NBCB(2)=0
            FB(1) = SSFND1
            FB(2) = SSFEND
            IF(I1SS.GT.IX) GO TO 1622
            I1SS  = IX
            NBCB(1)=2
            CURVE(1)=0.
 1622       NISS  = I-I1SS+1
            IF(NISS.EQ.5) GO TO 1624
            CALL BFAC(ZB(I1SS),RB(I1SS),ANG(I1SS),CURSS,NISS)
            GO TO 1626
 1624       CALL SS5PTC(ZB(I1SS),RB(I1SS),CURSS(NISS))
 1626       PHI1(M)=ANG(I)
            CURV(M)=CURSS(NISS)
            GO TO 164
  163 PHI1(M)=ANG(I)
      CURV(M)=CURVB(I)
      IF(I.NE.IA .OR. I.EQ.1) GO TO 164
      PHI1(M)=.5*(ANG(I)+ANGSAV)
      CURV(M)=.5*(CURVB(I)+CURSAV)
      IF(ISTAG.NE.1 .OR. CURVD.LE.0.) GO TO 164
      CURV(M)=CURVD
C     FBASTG= FRACTION OF BOUNDARY ANGLE AT STAGNATION POINT,CASC OPTION
      IF(FBASTG.EQ.0.) GO TO 1632
      CALL STANO(M,L,UPPER)
      IF(UPPER) ANGD=ANGD-PI
 1632 PHI1(M)=.5*(ANGSAV+(1.-FBASTG)*ANG(I)+FBASTG*ANGD)
      IF(ISTAG.EQ.1 .AND. CURVD.GT.0.) CURV(M)=CURVD
```

```
  164 S1(M) = S1B(I)
      GO TO 168
C       INTERPOLATE CURVATURE AND LOCATION FOR  ISTAG=3  POINTS
  166 DR     = RB(I+1)-RB(I)
      DZ     = ZB(I+1)-ZB(I)
      CHD    = SQRT(DR*DR+DZ*DZ)
      CS     = DZ/CHD
      SN     = DR/CHD
      ACHD   = ATAN3(DR,DZ,ANG(I))
      F      = (CS*(Z(M)-ZB(I))+SN*(R(M)-RB(I)))/CHD
      IF(F.GT.1. .OR. F.LT.0.) CALL ERROR1
      G      = 1.-F
      YPA    = TAN(ANG(I)-ACHD)
      YPB    = TAN(ANG(I+1)-ACHD)
      CALL BFI
      RM     = RB(I)+RM
      ZM     = ZB(I)+ZM
C     GUARD AGAINST UNREASONABLE INTERPOLATION
C     FIRST CHECK FOR EXISTENCE OF (M-1) AND (M+1)
C     THEN LIMIT THE POINT RELOCATION TO 25PC OF DIST. TO THESE NEIGHBOR
      F      = 1.
      CALL STAND(M,L,UPPER)
      IF(M.LE.MLB(L) .OR. UPPER) GO TO 167
      DM     = (R(M)-RM)**2 + (Z(M)-ZM)**2
      D1     = (R(M)-R(M-1))**2 + (Z(M)-Z(M-1))**2
      D2     = (R(M)-R(M+1))**2 + (Z(M)-Z(M+1))**2
      IF(DM.LT..01*D1) GO TO 167
      F      = AMIN1(1.,.25*SQRT(AMIN1(D1,D2)/DM))
  167 R(M)   = F*RM+(1.-F)*R(M)
      Z(M)   = F*ZM+(1.-F)*Z(M)
      PHI1(M)=ACHD+ANGM
      CURV(M)=CURVM
      S1(M)  = S1B(I)+S1M
C 168 IF(I.GE.IB) GO TO 170
  168 IF(PDUM(1).LE.0.) GO TO 1690
      IF(PDUM(1).EQ.1.) GO TO 1680
      IF(PDUM(1).EQ.2..AND. BI(I).LT.0.) GO TO 1680
      IF(PDUM(1).EQ.4. .AND. ISTAG.NE.0) GO TO 1680
      XJ2    = J2
      IF(PDUM(1).GE.5. .AND. XJ2.GE.PDUM(8) .AND. PDUM(1).GE.XJ2)
     1   GO TO 1680
      GO TO 1690
 1680 WRITE(6,1161) I,M,ISTAG,Z(M),R(M),PHI1(M),CURV(M),CURVB(I),B(I)
 1159 FORMAT(1H1)
 1160 FORMAT (12H    I M ISTAG,5X,1HZ,9X,1HR,4X,4HPHI1,4X,4HCURV,3X,
     1 5HCURVB,9X,1HB,5H   J=,I3)
 1161 FORMAT (1X,I3,I4,I2,2F10.5,F8.4,2F8.5,F10.3)
 1690 IF(I.GE.IB) GO TO 170
      M      = MD
      GO TO 161

C     INDEX TO NEXT STREAMLINE SEGMENT
  170 IF(MD) 172,180,172
  172 IA     = IB
      MA     = M
C     I      = IB
      ANGSAV= ANG(I)
      CURSAV= CURVB(I)
```

266

```
C      (TRANSFER TO 126 RATHER THAN 120 SINCE 1ST POINT, I=IA=IB, IS SAVE
       GO TO 126

C     STREAMLINE  J2  HAS BEEN CURVE-FITTED,  INDEX J2 TO NEXT SL.
  180 J2DONE(J2)=1
       ANYJ2 = .TRUE.
       GO TO 187
C       END CONDITION INTERPOLATION NOT POSSIBLE, BYPASS THIS SL
  186 ALLJ2 = .FALSE.
  187 J2    = J2+1
       IF(J2.LE.NJ) GO TO 101

C     GO BACK FOR 2ND, 3RD PASS TO INTERPOLATE FOR CURVATURE AT PARTIAL S
       IF(.NOT.ALLJ2) GO TO 100

       RZONLY= .TRUE.
       RETURN

 1155 FORMAT (27H * SLC IS INTERCHANGING PTS,F11.5,1H,F10.5,6H   AND,
      1   F11.5,1H,F10.5,4H  J=I3,5H   I=2I2)
       END
```

```
*DECK SPC
      SUBROUTINE SPC
*SPC---          SONIC POINT CURVATURE                              -SPC-

       COMMON /CSS    / SSFML,SSEF,SSEANG,SSDF,SSFEND,SSFND1
      1                 ,SSDLE,A4FACT,BRLX,CURRLX,TSIC
        INTEGER          SSFML
        LOGICAL                  SSEF,        SSDF,    SSDLE
C       SSFML  = SUPERSONIC CURVATURE FORMULA NUMBER
C       SSEF   = SUPERSONIC ENTERING FLOW, T OR F
C       SSEANG= ENTERING FLOW ANGLE (DEGREES) FOR SSEF=T
C       SSDF   = SUPERSONIC DISCHARGE FLOW, T OR F
C       SSFEND= SUPERSONIC BEAM DOWNSTREAM END CONDITION, =0,1 FOR PARABOL
C       SSFND1= SUPERSONIC BEAM UPSTREAM END CONDITION, =0,1, FOR PARABOLA
C       SSDLE  = SS FLOW BELOW AND AFT OF LE PT, T OR F
C       A4FACT= CENTRAL POINT INFLUENCE COEFFICIENT FACTOR
C       BRLX   = B-RELAXATION FACTOR
C       CURRLX= CURVATURE RELAXATION FACTOR
C       TSIC   = NUMBER OF POINTS TO BE USED FOR TRANSONIC INTERPOLATION
C                OF CURVATURE
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                 LO,LESTA,  LDUM(8),
      *                 MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                 LEO,LEE, LRO,LRE,LRD
        DIMENSION        LIMITS(24)
        EQUIVALENCE      (LIMITS,LHO)
       COMMON /SLTAB / W(128),X2(128),SLCHN(128)
        INTEGER          SLCHN
C      STATION TABLE
C       INDEX- L=LO,LESTA
C       SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C       MCL    = SHARP CORNER INDICATOR (BLDTBS)
C       MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
       COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
      1                 TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
      1                 TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
      3                 VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
        LOGICAL          PRIM
        INTEGER TYPELB,TYPEUB
        DIMENSION        SCHOKE(1)
        EQUIVALENCE      (SCHOKE,DWDV)
        DIMENSION        SLSWI(1)
        EQUIVALENCE      (SLSWI,VCL)
C       SLSWI  = SONIC LINE/SHOCK WAVE INDICATOR

       COMMON /CB     / B(300)
       COMMON /CCURV  / CURV(300)
       COMMON /CIDEX  / M,J,MU,MD,ISTAG
       COMMON /CS2    / S2(300)

C      RECOMPUTE NEAR SONIC PT CURVATURES BY LINEAR INTERPOLATION

C      BEGIN LOOP THROUGH STATIONS
       IF(TSIC.EQ.0.) RETURN
       L      = LO

C       CHECK FOR STAGNATION BOUNDARY POINT
   30  MA     = MLB(L)
```

268

```
      M      = MA
      CALL GETIX
      IF(ISTAG.NE.1) GO TO 40
      MA     = MA+1
   40 MB     = MUB(L)
      M      = MB
      CALL GETIX
      IF(ISTAG.NE.1) GO TO 50
      MB     = MB-1

C     LOCATE SONIC POINT
   50 IF(SLSWI(L).EQ.0.) GO TO 140
      M      = MA+1
   60 IF((B(M)*B(M-1)).GE.0.) GO TO 65
      CALL GETIX
      IF(W(J).NE.0.) GO TO 70
   65 M      = M+1
      IF(M.GT.MB) GO TO 140
      GO TO 60

C     F      = FRACTIONAL DISTANCE TO SONIC LINE ABOVE PT (M-1)
   70 F      = B(M-1)/(B(M-1)-B(M))

C     CALCULATION - INTERPOLATION JUNCTURE POINTS
      DFX    = AMIN1(TSIC,AMIN1(FLOAT(M-1-MA)+F,FLOAT(MB-M+1)-F))
      FX1    = F-DFX
      FX2    = F+DFX
      MX1    = M
      MX2    = M
   80 IF(FX1.GE.0. .OR. (MX1-1).LE.MA) GO TO 90
      MX1    = MX1-1
      FX1    = FX1+1.
      GO TO 80
   90 IF(FX2.LE.1. .OR. MX2.GE.MB) GO TO 100
      MX2    = MX2+1
      FX2    = FX2-1.
      GO TO 90
  100 SX1    = S2(MX1-1)+FX1*(S2(MX1)-S2(MX1-1))
      SX2    = S2(MX2-1)+FX2*(S2(MX2)-S2(MX2-1))

C     CALCULATE LINEAR VARIATION OF CURVATURE BET JUNCTURE PTS
      CX1    = CURV(MX1-1)+FX1*(CURV(MX1)-CURV(MX1-1))
      CX2    = CURV(MX2-1)+FX2*(CURV(MX2)-CURV(MX2-1))
      MX     = MX1
  120 IF(MX.GE.MX2) GO TO 65
      CURV(MX)=(CX1*(SX2-S2(MX))+CX2*(S2(MX)-SX1))/(SX2-SX1)
      MX     = MX+1
      GO TO 120

C     INDEX TO THE NEXT STATION
  140 L      = L+LNEXT(L)
      IF(L.LT.LESTA) GO TO 50

      RETURN
      END
```

```
*DECK SS5PTC
      SUBROUTINE SS5PTC(XX,YY,CURV)
*SS5PTC          SUPERSONIC 5-PT CURVATURE FORMULA                -SS5PTC-
       DIMENSION         XX(1),YY(1)

C     INPUT-
C      XX,YY = COORDINATES OF FIVE POINTS(I=1-5)

C     OUTPUT-
C      CURV  = CURVATURE OF THE LAST POINT (I=5)

      COMMON /CSS5PT/ X(4),Y(4), X21,X31,X32,X41,X42,X43, A0,A1,A2,A3,A4

      ALP    = ATAN2(YY(5)-YY(1),XX(5)-XX(1))
      SN     = SIN(ALP)
      CS     = COS(ALP)
C     X(0)   = 0.
      DO 60 I=1,4
      X(I)   = (XX(I+1)-XX(1))*CS+(YY(I+1)-YY(1))*SN
   60 Y(I)   = (YY(I+1)-YY(1))*CS-(XX(I+1)-XX(1))*SN
      CALL SS5PT
      D2YDX2= Y(1)*A1+Y(2)*A2+Y(3)*A3+Y(4)*A4
      CURV   = -D2YDX2
      RETURN
      END
```

```
*DECK STTOFI
      SUBROUTINE STTOFI(L1,MD1)
*STTOFI          ADJUST THE STATION-TABLE POINTERS          -STTOFI-
C                TO THE FIELD-TABLE UPWARD BY  MD1

C     INPUT-
C      L1     = FIRST STATION FOR WHICH POINTERS  MUB(L),MLB(L)  MUST BE A
C      MD1    = INCREMENT TO BE ADDED TO MLB(L) AND MUB(L).
C               MUB(L),MLB(L)  POINT TO THE FIELD-TABLE

       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
        DIMENSION      LIMITS(24)
        EQUIVALENCE    (LIMITS,LHO)
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
       COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
      1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
      1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
      3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
        LOGICAL        PRIM
        INTEGER TYPELB,TYPEUB
        DIMENSION      SCHOKE(1)
        EQUIVALENCE    (SCHOKE,DWDV)
       COMMON /CBITS / BITS,BLANK

       L      = L1
       MD     = MD1
       MUB(L)= MUB(L)+MD
       IF((MUB(L)-MLB(L)).LT.MAXOL) GO TO 60
       CALL ERROR1
    60 L      = L+LNEXT(L)
       IF(L.GE.LESTA) GO TO 900
       MLB(L)= MLB(L)+MD
       MUB(L)= MUB(L)+MD
       GO TO 60

   900 RETURN
       END
       OVERLAY(STC,4,0)
```

```
*DECK USECDM
      BLOCK DATA USECDM
*USECDM         REPLACE STCM USE CARDS
      COMMON /CA2    / A2(730)
      COMMON /CA2    / A2(768)
      COMMON /CA3    / A3(768)
      COMMON /CA4    / A4(768)
      COMMON /CA5    / A5(768)
      COMMON /CA6    / A6(768)
      COMMON /CA7    / A7(768)
      COMMON /CA8    / A8(768)
      END
```

```
*DECK ERROPM
      SUBROUTINE ERROPT
CERDUMPM          ERDUMP FOR STGM LINK


C     TABLE OF INDEX LIMITS
      COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
     *                LO,LESTA,  LDUM(8),
     *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
     *                LEO,LEE, LRO,LRE,LRD
       DIMENSION      LIMITS(24)
       EQUIVALENCE    (LIMITS,LHO)
C     STREAMLINE TABLE
      COMMON /SLTAB / W(128),X2(128),SLCHN(128)
       INTEGER SLCHN
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
      COMMON /CHDATA/ XL(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
     1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),SILB(1),
     1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),SIUB(1),
     3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
       LOGICAL        PRIM
       INTEGER TYPELB,TYPEUB
       DIMENSION      SCHOKE(1)
       EQUIVALENCE    (SCHOKE,DWDV)
      COMMON /CA2    / A2(300)
      COMMON /CA3    / A3(300)
      COMMON /CA4    / A4(300)
      COMMON /CA5    / A5(300)
      COMMON /CA6    / A6(300)
      COMMON /CA7    / A7(300)
      COMMON /CA8    / A8(300)
      COMMON /CB     / B(300)
      COMMON /CCURV  / CURV(300)
      COMMON /CDS2   / DS2(300)
      COMMON /CDDS2  / DDS2
      COMMON /CFB    / L,MA,MB,PLB,PUB,WF,CHOKE,SUBSON, NK,PLBC,PUBC,
     1                 XCHOKE, TAREA,VMBC, WRQST,WCALC, QV(8),QVP(8),
     *                 JSUM,VMLBSQ
       LOGICAL                                CHOKE,SUBSON
      COMMON /CIDEX/ C1(5)
      COMMON /CIDEXR/ C2(25)
      COMMON /CPHI1  / PHI1(300)
      COMMON /CR     / R(300)
      COMMON /CRHS   / RHS(300)
      COMMON /CS1    / S1(300)
      COMMON /CS2    / S2(300)
      COMMON /CTABPR/ I1TAB
      COMMON /CTOLRL/ C3(12)
      COMMON /CVM    / VM(300)
      COMMON /CZ     / Z(300)

      CALL TABPRT(3HCFB,L,33,4)
      CALL TABPRT (5HCIDEX,C1,5,5)
      CALL TABPRT (6HCIDEXR,C2,25,5)
```

274

```
      CALL TABPRT (6HCTOLRL,C3,6,6)
      IITAB  = LO
      CALL TABPRT (6HSTATAB,X1,LESTA,5)
      CALL JMSPRT
      CALL TABPRT (2HS1,S1,NM,10)
      CALL TABPRT (2HS2,S2,NM,10)
      CALL TABPRT (1HZ,Z,NM,10)
      CALL TABPRT (1HR,R,NM,10)
      CALL TABPRT(4HCURV,CURV,NM,10)
      CALL TABPRT (2HVM,VM,NM,10)

      WRITE (6,1000)
      DO 100 I=1,NM
      WRITE (6,1001) I,B(I),A2(I),A3(I),A4(I),A5(I),A6(I),A7(I),A8(I),
     1                DS2(I),RHS(I)
  100 CONTINUE
      WRITE (6,1002) DDS2
 1000 FORMAT (4H1  M,11X,1HB,10X,2HA2,10X,2HA3,10X,2HA4,10X,2HA5,10X,
     1         2HA6,10X,2HA7,10X,2HA8,9X,3HDS2,9X,3HRHS)
 1001 FORMAT (1H ,I3,8F12.3,2F12.6)
 1002 FORMAT(///8H  DS2MX=,F12.6)
      LSTOP = 5
      GO TO (999,999) , LSTOP
  999 RETURN
      END
```

```
*DECK MCOEF
      SUBROUTINE MCOEF
*MCOEF            MATRIX COEFFICIENT                              -MCOEF-

C     INPUT-
C      W(J)   = SL FLOW
C      S1(M)  = DISTANCE ALONG STREAMLINES
C      B(M)   = COEFFICIENT OF THE CURVATURE TERM
C      STATION TABLE

C     OUTPUT-
C      A1(M),A2(M),..A8(M) = MATRIX COEFICIENT ARRAYS        M=1,NM

C           STAR ARRANGEMENT IS -
C                                                  A8
C                        A1       A2       A3     A4       A5       A6
C                                                  A7

C      NOTE -  A4 IS ALWAYS NEGATIVE EXCEPT FOR THE FIRST OF DOUBLE POINT
C              THEN A4(M)=1., A8(M)=-1.

       COMMON /BENDIN/ NBCIN(2),ACF(2)
       COMMON /CA2    / A2(300)
       COMMON /CA3    / A3(300)
       COMMON /CA4    / A4(300)
       COMMON /CA5    / A5(300)
       COMMON /CA6    / A6(300)
       COMMON /CA7    / A7(300)
       COMMON /CA8    / A8(300)
        DIMENSION       A0(300),A1(300)
        EQUIVALENCE     (A0,A6),(A1,A5)
       COMMON /CATM   / NX,XDIM,G(25)
       COMMON /CB     / B(300)
       COMMON /CBITS  / BITS,BLANK
       COMMON /CCUBE  / NBC(2),C1(2),C2(2),FEND(2)
       COMMON /CCURV  / CURV(300)
       COMMON /CFB    / L,MA,MB,DFB(30)
       COMMON /CFFINC/ GFF(6)
       COMMON /CFRFIN/ ATINF
       COMMON /CIDEX  / M,J,MU,MD,ISTAG
       COMMON /CMAXIT/ MAXIT,MAJCTR,GREFIN
       COMMON /CPI    / PI,TWOPI,PIQ2,PIQ4,TODEG,TORAD
       COMMON /CPRINT/ PRTES2,PRTB,PRTA,PREFIN,PREFN2,SSONIC,PDUM(10)
       COMMON /CR     / R(300)
       COMMON /CRHS   / RHS(300)
       COMMON /CS1    / S1(300)
       COMMON /CSS    / SSFML,SSEF,SSEANG,SSDF,SSFEND,SSFND1
      1                 ,SSDLE,A4FACT,BRLX,CURRLX,TSIC,RHOC,RHOCSS
        INTEGER         SSFML
        LOGICAL               SSEF,       SSDF,   SSDLE
C      SSFML = SUPERSONIC CURVATURE FORMULA NUMBER
C      SSEF  = SUPERSONIC ENTERING FLOW, T OR F
C      SSEANG= ENTERING FLOW ANGLE (DEGREES) FOR SSEF=T
C      SSDF  = SUPERSONIC DISCHARGE FLOW, T OR F
C      SSFEND= SUPERSONIC BEAM DOWNSTREAM END CONDITION, =0,1 FOR PARABOL
C      SSFND1= SUPERSONIC BEAM UPSTREAM END CONDITION, =0,1, FOR PARABOLA
C      SSDLE = SS FLOW BELOW AND AFT OF LE PT, T OR F
```

```
C      A4FACT= CENTRAL POINT INFLUENCE COEFFICIENT FACTOR
C      BRLX   = B-RELAXATION FACTOR
C      CURRLX= CURVATURE RELAXATION FACTOR
       COMMON /CVM    / VM(300)
       COMMON /CXG    / XO,X(6)
       COMMON /ALLCOM/ MACHA,PSA,TSA,PTA,TTA, AXIA,RGA,GAMA,
      1                MACHC,PSC,TSC,PTC,TTC, AXIC,RGC,GAMC,
      2                DAXIT,SCALEA,TTE,CHOTST
        REAL           MACHA(1),MACHC
        LOGICAL        AXIA,AXIC
        LOGICAL        CHOTST
       COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
      *                LO,LESTA,  LDUM(8),
      *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
      *                LEO,LEE, LRO,LRE,LRD
        DIMENSION      LIMITS(24)
        EQUIVALENCE    (LIMITS,LHO)
       COMMON /SLTAB / W(128),X2(128),SLCHN(128)
        INTEGER SLCHN
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
       COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
      1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
      1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
      3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
        LOGICAL        PRIM
        INTEGER TYPELB,TYPEUB
        DIMENSION      SCHOKE(1)
        EQUIVALENCE    (SCHOKE,DWDV)


       INTEGER         FIELD,FREE,FARFLD,PRES,OLBC
       LOGICAL         SLBDY,SUBDY

       DATA FIELD/5HFIELD/
       DATA FREE/4HFREE/, FARFLD/6HFARFLD/, PRES/4HPRES/, OLBC/4HOLBC/


C      BEGIN LOOP THROUGH THE STATIONS
       L      = LO

C      BEGIN LOOP ACROSS THE STREAMLINES
  800  MA     = MLB(L)
       MB     = MUB(L)
       M      = MA
  810  A2(M)  = 0.
       A3(M)  = 0.
       A4(M)  = 0.
       A5(M)  = 0.
       A6(M)  = 0.
       A7(M)  = 0.
       A8(M)  = 0.
       MCENTR= M

C      INITIALIZE /CCUBE/ FOR CUFITR
```

277

```
      C1(1) = 0.
      C1(2) = 0.
      C2(1) = 0.
      C2(2) = 0.


C     CHECK FOR SPECIAL (FREE,PRES, OR FARFLD) BOUNDARY
      SLBDY = .FALSE.
      SUBDY = .FALSE.
      IF(M.NE.MA) GO TO 824
      IF(TYPELB(L).EQ.FREE .OR.
     1   TYPELB(L).EQ.FARFLD .OR.
     3   TYPELB(L).EQ.FIELD .OR.
     *   TYPELB(L).EQ.OLBC .OR.
     2   TYPELB(L).EQ.PRES) SLBDY=.TRUE.
      IF(.NOT.SLBDY) GO TO 825
  824 IF(M.NE.MB) GO TO 826
      IF(TYPEUB(L).EQ.FREE .OR.
     1   TYPEUB(L).EQ.FARFLD .OR.
     3   TYPEUB(L).EQ.FIELD .OR.
     4   TYPEUB(L).EQ.OLBC .OR.
     2   TYPEUB(L).EQ.PRES) SUBDY=.TRUE.
      IF(SUBDY) GO TO 826

C     SOLID WALL BOUNDARY
  825 A4(M) = -1.
      GO TO 980

C     INTERIOR POINT
C     BUILD X-TABLE OF DISTANCES TO NEIGHBORING POINTS ALONG THE STREAMLI
C       POINTS WITH ISTAG=3 ARE TO BE OMITTED.
C       SPECIAL END CONDITIONS ARE TO BE UTILIZED IF THE X-TABLE IS TERMI
C       BY A STAGNATION POINT
  826 CALL GETIX
      JCENTR= J
      ISTAGC= ISTAG
      X(4)  = S1(M)
      IC1   = 4
      IC2   = 4
      NBC(1)= 2
      NBC(2)= 2
      C2(1) = 0.
      C2(2) = 0.
      MDOWN = MD

  831 M     = MU
      IF(M.EQ.0) GO TO 850
      CALL GETIX
      IF(ISTAG.EQ.3) GO TO 831
      X(3)  = S1(M)
      IC1   = 3
      IF(ISTAG.NE.0) GO TO 850
      IF( SSFML.LT.0 .AND. B(MCENTR).GE.0. .AND. PDUM(12).EQ.(-1.) )
     1       GO TO 850

  841 M     = MU
      IF(M.EQ.0) GO TO 850
      CALL GETIX
```

```
      IF(ISTAG.EQ.3) GO TO 841
      X(2)   = S1(M)
      IC1    = 2

  846 IF( B(MCENTR).GT.0. ) GO TO 850
        IF(IABS(SSFML).EQ.1 ) GO TO 850
      M      = MU
      IF(M.EQ.0) GO TO 850
      CALL GETIX
      IF(ISTAG.EQ.3) GO TO 846
      X(1)   = S1(M)
      IC1    = 1

  848 IF( SSFML.NE.3 ) GO TO 850
      IC1    = 2
      M      = MU
      IF( M.EQ.0 ) GO TO 850
      CALL GETIX
      IF( ISTAG.EQ.3 ) GO TO 848
      XO     = S1(M)
      IC1    = 0

C     UPSTREAM STREAMLINE END CONDITION
  850 IF(MU) 854,852,854
  852 NBC(1)= NBCIN(1)
      FEND(1)=ACF(1)

C     DOWNSTREAM POINTS, BYPASS FOR SUPERSONIC FLOW
  854 IF(B(MCENTR).LE.0.) GO TO 874
      MD     = MDOWN
  856 M      = MD
      IF(M.EQ.0) GO TO 870
      CALL GETIX
      IF(ISTAG.EQ.3) GO TO 856
      X(5)   = S1(M)
      IC2    = 5
      IF(ISTAG.NE.0) GO TO 865
      IF( SSFML.LT.0 .AND. PDUM(12).EQ.(-1.) ) GO TO 865

  861 M      = MD
      IF(M.EQ.0) GO TO 870
      CALL GETIX
C     IF(B(M).LE.0. .AND. B(MU).LE.0.) GO TO 874
      IF(ISTAG.EQ.3) GO TO 861
      X(6)   = S1(M)
      IC2    = 6

C     SPECIAL DOWNSTREAM END CONDITIONS FOR LEADING EDGE STAGNATION POINT
  865 IF(ISTAG.NE.1) GO TO 870
      NBC(2)= 4
      C1(2) = CURV(M)
      FEND(2)=1.
      IF(ABS(PDUM(5)).GE.5.) FEND(2)=0.

C     DOWNSTREAM STREAMLINE END CONDITIONS
  870 IF(MD) 878,872,878
  872 NBC(2)= NBCIN(2)
      FEND(2)=ACF(2)
```

```
      GO TO 878

C     BOUNDARY CONDITION ON 4-POINT SUPERSONIC BEAM-CURVATURE FORMULA
  874 FEND(2)=SSFEND
      FEND(1)=SSFND1
      NBC(2)= 0
      NBC(1)= 0

C     SUPERSONIC 5-PT FORMULA
      IF( SSFML.NE.3 .OR. IC1.NE.0 ) GO TO 878
      CALL SS5PTI( XO,G )
      GO TO 900

C     CALL CUBER TO OBTAIN SECOND ORDER DIFFERENCE FORMULA, D2(DN)/D(S1)2
C      ANSWERS ARE STORED IN  G(IG,JG), JG=1,IC2-IC1+1, IG=MID POINT
  878 NIC    = IC2-IC1+1
      IF(ISTAGC.EQ.3) GO TO 880
      IF(NIC.LE.2) GO TO 908
      CALL CUBERS(X(IC1),NIC)
      GO TO 900

C     CALL CUFITR FOR INFLUENCE COEFFICIENTS, DS2(3)=F(DS2(1),DS2(2),DS(4
C       FOR INFIELD BOUNDARY POINT(ISTAG=3)
  880 CALL CUFITR(X(IC1),NIC,5-IC1)

C****DEFINE ALL COEFFICIENTS OF THE EQUATION FOR FIELD POINT M
C      IG     = 4-IC1+1
C      JG     = IC-IC1+1
C      IJG    = (JG-1)*5 + IG
C      IJG    = CENTER POINT INDEX IN G-ARRAY
  900 IJG    = 25-IC1*6
      IF(PDUM(5).LE.0.) GO TO 907
      IF(ISTAGC.NE.3 .AND. PDUM(5).EQ.3.) GO TO 907
      IF(PDUM(5).GE.4. .AND. NBC(2).NE.4) GO TO 907
      WRITE (6,1907) JCENTR,MCENTR,IC1,IC2,IJG
 1907 FORMAT(//3H J=I3,9H  MCENTR=I3,7H   IC1=I3,7H   IC2=I3,6H  IJG=I3)
      CALL TABPRT(1HX,X(IC1),NIC,5)
      CALL TABPRT(5HCCUBE,NBC,8,8)
      CALL TABPRT(1HG,G,25,5)
  907 CONTINUE

C     CHECK FOR INFIELD BOUNDARY POINT OR SPECIAL BOUNDARY
  908 IF(.NOT.SLBDY .AND. .NOT.SUBDY .AND. ISTAGC.NE.3) GO TO 910
      M      = MCENTR
      GO TO 926

C     FIRST POINT OF A DOUBLE SL, CHECK  W(JCENTR+1)
  910 M      = MCENTR+1
      CALL GETIX
      IF(W(J).NE.0.) GO TO 915
      M      = MCENTR
      J      = JCENTR
      GO TO 926

C     POINTS 7, 8, AND 4
  915 JP     = J
      MP     = M
      JM     = JCENTR
```

```
      M      = MCENTR-1
      IF(W(JCENTR).NE.0.) GO TO 920
      CALL GETIX
      JM     = J
      M      = MCENTR-2
  920 CALL GETIX
      MM1    = M
      JM1    = J
      M      = MCENTR
      J      = JCENTR

      A7(M) = 1./(W(JM)-W(JM1))
      A8(M) = 1./(W(JP)-W(J))
      A4(M) = -A7(M)-A8(M)
      IF(.NOT.AXIA) GO TO 926
      A4(M) = TWOPI*R(M)*A4(M)
      A7(M) = TWOPI*R(MM1)*A7(M)
      A8(M) = TWOPI*R(M+1)*A8(M)

C     POINTS 1, 2, 3, 4, 5, AND 6
  926 IF(NIC.LE.2) GO TO 938
      BUSE   = RHOC*B(M)
      IF(B(M).LT.0.) BUSE=RHOCSS*B(M)
      IF(IC1.NE.0 ) GO TO 930
      A0(M) = BUSE*G(5)
      GO TO 931
  930 GO TO (931,932,933,934),IC1
  931 A1(M) = BUSE*G(IJG-15)
  932 A2(M) = BUSE*G(IJG-10)
  933 A3(M) = BUSE*G(IJG-5)
  934 A4(M) = BUSE*G(IJG)+A4(M)
      IF(IC2-5) 938,935,936
  936 A6(M) = BUSE*G(IJG+10)
  935 A5(M) = BUSE*G(IJG+5)

C     MODIFY INFLUENCE COEFFICIENTS TO ACCOMMODATE DOUBLE STREAMLINE
C       MT     = TRUE POINT
C       MX     = DUMMY POINT
C       MX IS THE FIRST POINT, EXCEPT FOR CASC PROG WITH UPPER OLBC.
C       THEN MX IS THE SECOND POINT.
  938 IF(W(J).NE.0. .OR. SLBDY) GO TO 940
      MT     = M
      MX     = M-1
      IF(TYPEUB(L).NE.OLBC) GO TO 19392
      MT     = M-1
      MX     = M
19392 A2(MT)= A2(M)+A2(M-1)
      A3(MT)= A3(M)+A3(M-1)
      A4(MT)= A4(M)+A4(M-1)
      A5(MT)= A5(M)+A5(M-1)
      A6(MT)= A6(M)+A6(M-1)
      IF(MX.NE.M) GO TO 19394
C       MX=M  AND  MT=M-1
      A7(M-1)=A7(M)
      A8(M-1)=A8(M)
      RHSV   = RHS(M-1)
      RHS(M-1)=RHS(M)
      A7(M) = -1.
```

```
      A8(M) = 0.
      RHS(M)= -RHSV
      GO TO 19396
19394 A7(MX)= 0.
      A8(MX)= -1.
19396 A2(MX)= 0.
      A3(MX)= 0.
      A4(MX)= 1.
      A5(MX)= 0.
      A6(MX)= 0.

C     FREE, PRESSURE AND FAR-FIELD BOUNDARIES
C       LOWER BOUNDARY
  940 IF(ISTAGC.EQ.3) GO TO 980
      IF(.NOT.SLBDY) GO TO 950
      IF(.NOT.AXIA) GO TO 942
      A4(M) = A4(M)-TWOPI*R(M)
      A8(M) = TWOPI*R(M+1)
      GO TO 980
  942 A4(M) = A4(M)-1.
      A8(M) = 1.
      IF(TYPELB(L).NE.FARFLD) GO TO 980
C     STAREA= STREAM TUBE AREA
      STAREA= R(M+1)-R(M)
      IF(AXIA) STAREA=PI*(R(M)+R(M+1))*STAREA
  945 CALL FFINC
      VQATSQ= VM(M)*VM(M)/(ATINF*ATINF)
      BETA  = 1.-VQATSQ/(1.-.2*VQATSQ)
      IF(BETA.GT.0.) GO TO 947
      WRITE (6,1946) M
      CALL ERROR1
 1946 FORMAT(76H ***  SORRY - SUPERSONIC VELOCITY ENCOUNTERED ON FAR FIE
     1LD BOUNDARY AT POINT, I5, 9H  (MCOEF))
  947 BETA  = SQRT(BETA)
      BA    = BETA*STAREA
      A2(M) = A2(M)-BA*GFF(2)
      A3(M) = A3(M)-BA*GFF(3)
      A4(M) = A4(M)-BA*GFF(4)
      A5(M) = A5(M)-BA*GFF(5)
      A6(M) = A6(M)-BA*GFF(6)
      GO TO 980
C       UPPER BOUNDARY
  950 IF(.NOT.SUBDY) GO TO 980
      IF(AXIA) GO TO 964
      A4(M) = A4(M)-1.
      A7(M) = 1.
      GO TO 966
  964 A4(M) = A4(M)-TWOPI*R(M)
      A7(M) = TWOPI*R(M-1)
  966 IF(TYPEUB(L).NE.FARFLD) GO TO 980
      STAREA= R(M)-R(M-1)
      IF(AXIA) STAREA=PI*(R(M)+R(M-1))*STAREA
      GO TO 945

  980 M     = MCENTR+1
      IF(M.LE.MB) GO TO 810
C.....END LOOP ACROSS THE STREAMLINES
```

```
C     INDEX TO NEXT STATION
      L      = L+LNEXT(L)
      IF(L.LT.LESTA) GO TO 800
C.....END LOOP THROUGH THE STATIONS

      IF(PDUM(3).EQ.0.) GO TO 990
      WRITE (6,1000)
      DO 100 I=1,NM
      WRITE (6,1001) I,B(I),A2(I),A3(I),A4(I),A5(I),A6(I),A7(I),A8(I),
     1                          RHS(I)
  100 CONTINUE
  990 RETURN
 1000 FORMAT (4H1  M,11X,1HB,10X,2HA2,10X,2HA3,10X,2HA4,10X,2HA5,10X,
     1         2HA6,10X,2HA7,10X,2HA8,21X,3HRHS)
 1001 FORMAT (1H ,I3,8F12.3,12X,F12.6)
      END
```

```
*DECK ATDMRS
      SUBROUTINE ATDMRS
*ATDMRS          AUGUMENTED TRIDIAGONAL MATRIX REDUCTION      -ATDMRS-
C                SMALL MATRIX VERSION

C     GIVEN THE MATRIX EQUATION  AX=BY,
C      FIND  G  SO THAT  X=GY,
C      NOTE  X  AND  Y  ARE VECTORS.

C     INPUT-
C      A      - TRIDIAGONAL COEFFICIENT MATRIX OF X
C      B      = TRIDIAGONAL COEFFICIENT MATRIX OF Y (STORED IN G-ARRAY)
C               (OTHER OFF-DIAGONAL ELEMENTS MUST BE INITIALIZED TO ZERO)
C      IDIM   - FIRST SUBSCRIPT DIMENSION OF MATRIX  B  AND  G
C      N      - ORDER OF MATRACIES

C     ORDER OF STORAGE IS ILLUSTRATED BY-
C       A(2,1)  A(3,1) (A(1,1))          B(1,1)  B(1,2)
C       A(1,2)  A(2,2)  A(3,2)           B(2,1)  B(2,2)  B(2,3)
C               A(1,3)  A(2,3)  A(3,3)           B(3,2)  B(3,3)  B(3,4)
C             (A(3,4)) A(1,4)  A(2,4)                   B(4,3)  B(4,4)

C      OFF DIAGONAL ELEMENTS OF B MUST BE SET TO ZERO

C     OUTPUT-
C      G      - INVERSE(A) * B

       COMMON /ERASE / A(3,100), DUM(500)
       COMMON /CATM  / N,IDIM,G(25)

C*** FORWARD REDUCTION
       A(3,1)= A(3,1)/A(2,1)
       G(1)  = G(1)/A(2,1)
       G(IDIM+1)=G(IDIM+1)/A(2,1)
       I     = 2

C       SPECIAL LOGIC FOR INCLUDING  A(4,1)  WHICH IS STORED IN  A(1,1)
       A(1,1)= A(1,1)/A(2,1)
       QA2I  = 1./(A(2,2)-A(1,2)*A(3,1))
       A(3,2)= QA2I*(A(3,2)-A(1,2)*A(1,1))
       GO TO 97

    90 QA2I  = 1./(A(2,I)-A(1,I)*A(3,I-1))
    95 A(3,I)= QA2I*A(3,I)
    97 J     = 1
       IJ    = I
   120 G(IJ) = QA2I*(G(IJ)-A(1,I)*G(IJ-1))
       IF(J-I) 140,140,160
   140 IF(J-N) 150,160,160
   150 J     = J+1
       IJ    = IJ+IDIM
       GO TO 120
   160 IF(I-N) 170,180,170
   170 I     = I+1

C       SPECIAL LOGIC FOR INCLUDING A(N,N-2) WHICH IS STORED IN A(3,N)
       IF(I-N) 90,172,172
   172 A(1,I)= A(1,I)-A(3,I)*A(3,I-2)
```

```
      J      = 1
      IJ     = I
  178 G(IJ) = G(IJ)-A(3,I)*G(IJ-2)
  179 J      = J+1
      IJ     = IJ+IDIM
      IF(J-I)178,90,90


C*** BACK SUBSTITUTION
  180 I      = I-1
C     IF(I) 900,900,190
  190 J      = 1
      IJ     = I

C      SPECIAL LOGIC FOR INCLUDING  A(4,1)  WHICH IS STORED IN  A(1,1)
  192 IF(I-1) 900,195,200
  195 G(IJ) = G(IJ)-A(1,1)*G(IJ+2)

  200 G(IJ) = G(IJ)-A(3,I)*G(IJ+1)
      IF(J.EQ.N) GO TO 180
      J      = J+1
      IJ     = IJ+IDIM
      GO TO 192

  900 RETURN
      END
```

```
*DECK CUBE
      SUBROUTINE CUBE(X,Y,NN,B)
*CUBE--          FIT A SERIES OF CUBICS TO POINTS                    -CUBE-
*                END CONDITIONS ARE ARBITRARY
      DIMENSION X(10),Y(10),B(10)

C     ON ENTRY -
C      X,Y    = LISTS OF COORDINATES
C      N      = NO. OF POINTS (N.GE.2)

C       ALSO DEFINED ON ENTRY - IN COMMON/CCUBE/ -
C      NBC(L)= BOUNDARY CONDITION INDICATOR FOR LEFT(L=1) AND RIGHT(L=2)
C            = 0, 1, OR 2
C      YP(L) = FIRST DERIVATIVE  IF  NBC(L)=1
C      YPP(L)= SECOND DERIVATIVE IF  NBC(L)=2

C     ON RETURN-
C      B(I)   = FIRST  DERIVATIVE AT POINT  I  (I=1,N)

      COMMON /CCUBE / NBC(2),YP(2),YPP(2),FEND(2)
      COMMON /CCUBIC/ N,IA,IB
      COMMON /ERASE / A(3,266),DRASE(2)

      LOGICAL PARAB


C     INITIALIZE
      N      = NN
      IA     = 2
      IB     = N-1
      DX1    = X(2)-X(1)
      DY1    = Y(2)-Y(1)
      DXN    = X(N)-X(N-1)
      DYN    = Y(N)-Y(N-1)
C       NOTE -DXN- IS THE DELTA X FOR THE (N-1) INTERVAL.  DXNM1 WOULD BE
C       MORE PRECISE SYMBOL.

C     A STRAIGHT LINE IS USED FOR N=2 IF  NBC(1)=NBC(2)=0
      NBCS = NBC(1)+NBC(2)
      IF(N.GT.2 .OR. NBCS.GT.0) GO TO 80
      B(1)   = (Y(2)-Y(1))/(X(2)-X(1))
      B(2)   = B(1)
      GO TO 900

C     CHECK IF PARABOLA (F=0) SHOULD BE USED
   80 PARAB = (N.EQ.2 .AND. (NBC(1)*NBC(2)).EQ.0)  .OR.
     1         (N.EQ.3 .AND. NBCS.EQ.0)

C     NBC=01,  Y  AND  YP  SPECIFIED
C       LEFT END
  110 IF(NBC(1).NE.01) GO TO 120
      A(2,1)= 1.
      A(3,1)= 0.
      B(1)   = YP(1)
C       RIGHT END
  120 IF(NBC(2).NE.01) GO TO 210
      A(1,N)= 0.
      A(2,N)= 1.
```

```
      B(N)  = YP(2)

C     NBC=02,  Y  AND  YPP  SPECIFIED
C       LEFT END
  210 IF(NBC(1).NE.02) GO TO 220
      A(2,1)= 4.
      A(3,1)= 2.
      B(1)  = 6.*DY1/DX1 - YPP(1)*DX1
C       RIGHT END
  220 IF(NBC(2).NE.02) GO TO 310
      A(1,N)= 2.
      A(2,N)= 4.
      B(N)  = YPP(2)*DXN + 6.*DYN/DXN

C     NBC=0,   YPPP = F * YPPP(OF ADJACENT INTERVAL)
C       LEFT END
  310 IF(NBC(1).NE.0) GO TO 320
      A(2,1)= 1.
      A(3,1)= 1.
      B(1)  = 2.*DY1/DX1
      IF(PARAB) GO TO 320
      DX2   = X(3)-X(2)
      DY2   = Y(3)-Y(2)
      DX1DX2= DX1/DX2
      A(2,1)= A(2,1) + FEND(1)*DX1DX2
      A(3,1)= A(3,1) + FEND(1)*DX1DX2*(2.+DX1DX2)
      B(1)  = B(1) + FEND(1)*(3.*DY1+DY2*DX1DX2*DX1DX2)/DX2
C       RIGHT END
  320 IF(NBC(2).NE.0) GO TO 500
      A(1,N)= 1.
      A(2,N)= 1.
      B(N)  = 2.*DYN/DXN
      IF(PARAB) GO TO 500
      DXM   = X(N-1)-X(N-2)
      DYM   = Y(N-1)-Y(N-2)
      DXNDXM= DXN/DXM
      A(1,N)= A(1,N) + FEND(2)*DXNDXM*(2.+DXNDXM)
      A(2,N)= A(2,N) + FEND(2)*DXNDXM
      B(N)  = B(N) + FEND(2)*(3.*DYN+DYM*DXNDXM*DXNDXM)/DXM

  500 CALL CUBICS(X,Y,B)

  900 RETURN
      END
```

```
*DECK CCUBE
      BLOCK DATA CUBEBK
*CCUBE-          DATA FOR /CCUBE /                                  -CCUBE-
      COMMON /CCUBE / NBC(2),YP(2),YPP(2),FEND(2)
      DATA NBC,YP,YPP,FEND/2*0,6*0./
      END
```

```
*DECK CUBERS
      SUBROUTINE CUBERS(X,NN)
*CUBERS          YPP  IN TERMS OF  Y                              -CUBERS-
C                FOR CUBIC SPLINE EQUATIONS

C                SPECIAL SMALL MATRIX VERSION WITH END CONDITIONS FOR -STC

       DIMENSION      X(10)

C     ON ENTRY -
C      X      = LIST OF DISTANCES
C      NN     = NO. OF POINTS (N.GE.3)

C       ALSO DEFINED ON ENTRY - IN COMMON/CCUBE/ -
C      NBC(L)= BOUNDARY CONDITION INDICATOR FOR LEFT(L=1) AND RIGHT(L=2)
C            = 0 IF FEND(L) IS SPECIFIED
C            = 1 FOR  YP(L)=0.
C            = 2 FOR  YPP(L)=0.
C            = 4 FOR  YP(L)=-C1(L)*Y(L)   AND  YPPP(L)=FEND(L)*YPPP(NEXT
C      FEND(L)= END/NEXT TO END VALUE OF YPPP IF NBC(L)=0

C     ON RETURN-
C      G(I,J)- MATRIX DEFINED BY  C=GY  WHERE C IS A VECTOR OF SECOND DER

       COMMON /CATM  / N,IDIM,B(5,5)
       COMMON /CCUBE / NBC(2),C1(2),C2(2),FEND(2)
       COMMON /ERASE / A(3,266),DRASE(2)

C****DEFINE COEFFICIENT MATRICIES -A- AND -B-, WHERE  A*YPP=B*Y

C     INITIALIZE
       N      = NN
       F1     = FEND(1)
       F2     = FEND(2)
       IF(N-3) 60,65,70
    60 CALL ERROR1
    65 F1     = 0.
       F2     = 0.
    70 CALL SETM(2,0., A,15,  B,25)
       DX1    = X(2)-X(1)
       DX2    = X(3)-X(2)
       DXM    = X(N-1)-X(N-2)
       DXN    = X(N)-X(N-1)
C       NOTE -DXN- IS THE DELTA X FOR THE (N-1) INTERVAL.  DXNM1 WOULD BE
C       MORE PRECISE SYMBOL.
       IA     = 2
       IB     = N-1

C     NBC=01,  YP=0.
C       LEFT END
  110 IF(NBC(1).NE.01) GO TO 120
       A(2,1)= DX1+DX1
       A(3,1)= DX1
       B(1,2)= 6./DX1
       B(1,1)= -B(1,2)
C       RIGHT END
  120 IF(NBC(2).NE.01) GO TO 210
       A(1,N)= DXN
```

```
      A(2,N)= DXN+DXN
      B(N,N-1)=6./DXN
      B(N,N)= -B(N,N-1)

C     NBC=02,  YPP=0.
C       LEFT END
  210 IF(NBC(1).NE.02) GO TO 220
      A(2,1)= 1.
C       RIGHT END
  220 IF(NBC(2).NE.02) GO TO 310
      A(2,N)= 1.

C     NBC=0,   YPPP = F * YPPP(OF ADJACENT INTERVAL)
C       LEFT END
  310 IF(NBC(1).NE.0)  GO TO 320
      A(1,1)= F1*DX1
      A(2,1)= DX2
      A(3,1)= -DX2-A(1,1)
C       RIGHT END
  320 IF(NBC(2).NE.0) GO TO 410
      A(3,N)= F2*DXN
      A(2,N)= DXM
      A(1,N)= -DXM-A(3,N)

C     NBC=04,  YP=C1*Y  AND  YPPP=F*YPPP(NEXT TO END)
C       LEFT END
  410 IF(NBC(1).NE.04) GO TO 420
      CALL ERROR1
C       RIGHT END
  420 IF(NBC(2).NE.04) GO TO 500
      A(2,N)= 1.
      IB    = N-2
      ADXN  = C1(2)*DXN
      C1PAD = 1.+ADXN
      A(1,N-1)= DXM + F2*DXN*DXN/DXM*(3.+ADXN)/C1PAD
      A(2,N-1)= A(1,N-1)+DXM+3.*DXN*(2.+ADXN)/C1PAD
      B(N-1,N-2)=6./DXM
      B(N-1,N-1)=-6.*(1./DXM+C1(2)/C1PAD)

C     CUBIC RECURSION FORMULA BASED ON MATCHING  YP  AND  YPP
  500 IF(IB.LT.IA) GO TO 600
      DO 550 I=IA,IB
      A(1,I)=X(I)-X(I-1)
      A(3,I)=X(I+1)-X(I)
      A(2,I)=2.*(A(1,I)+A(3,I))
      B(I,I-1)=6./A(1,I)
      B(I,I+1)=6./A(3,I)
  550 B(I,I)=-B(I,I-1)-B(I,I+1)

C****DETERMINATION OF -G- BY MATRIX REDUCTION,  YPP=G*Y
  600 IDIM  = 5
      CALL ATDMRS

  900 RETURN
      END
```

290

```
*DECK CUBICS
      SUBROUTINE CUBICS(X,Y,B)
*CUBICS         SERIES OF CUBICS FIT TO COORDINATE POINTS    -CUBICS-
      DIMENSION X(100), Y(100), B(100)

C     INPUT-
C      X(I),Y(I)
C      A(1,I),A(2,I),A(3,I),B(I)   I=1,(IA-1)  AND  I=(IB+1),N  (I.E. B.C
C      IA,IB    RANGE IN WHICH THE COEFFICIENT MATRIX AND CONSTANT VECTOR
C               BE DEFINED BY EQUATIONS FOR MATCHING  YP  AND  YPP.
C      1,N      RANGE OF X,Y, AND B

C     OUTPUT
C      B(I)     SLOPE AT X(I)

      COMMON /CCUBIC/ N,IA,IB
      COMMON /ERASE / A(3,266), DRASE(2)

C     SET UP TRIDIAGONAL COEFFICIENT MATRIX  A  AND VECTOR  B.  ORDER OF
C      STORAGE IS ILLUSTRATED BY -
C         A(2,1)    A(3,1)                              B(1)
C         A(1,2)    A(2,2)    A(3,2)                    B(2)
C                   A(1,3)    A(2,3)    A(3,3)          B(3)
C                             A(1,4)    A(2,4)          B(4)

C      I      = POINTS AT WHICH YP AND YPP ARE MATCHED
C      IA,IB  = LIMITS OF I

      IF(IB.LT.IA) GO TO 100
      DO 70 I=IA,IB
      A(1,I)= X(I+1)-X(I)
      A(3,I)= X(I)-X(I-1)
      A(2,I)= 2.*(A(1,I)+A(3,I))
   70 B(I)  =3.*((Y(I+1)-Y(I))*A(3,I)/A(1,I)+(Y(I)-Y(I-1))*A(1,I)/A(3,I)
     1        )

C     ROUTINE  TDSEQ  - TRIDIAGONAL SIMULTANEOUS EQUATIONS
C      SOLUTION TO  AX=B.   ON RETURN SOLUTION VECTOR X IS STORED IN  B.
  100 A(3,1)= A(3,1)/A(2,1)
      B(1)  = B(1)/A(2,1)
      DO 150 I=2,N
      A(2,I)= A(2,I)-A(1,I)*A(3,I-1)
      A(3,I)= A(3,I)/A(2,I)
  150 B(I)  = (B(I)-A(1,I)*B(I-1)) / A(2,I)

      I      = N
  200 I      = I-1
      IF(I.LE.0) GO TO 900
      B(I)  = B(I)-A(3,I)*B(I+1)
      GO TO 200

  900 RETURN
      END
```

```
*DECK CUFIT
      SUBROUTINE CUFIT(X,Y,NPTS, NEW, XC,YC,NXC,ND, B)
*CUFIT-                                                          -CUFIT-
C            INTEGRATE, INTERPOLATE FOR COORDINATES, 1ST, OR, 2ND DERIVAT
C            BY A CUBIC SPLINE CURVE FIT

       LOGICAL                    NEW
      DIMENSION X(10),Y(10), XC(10),YC(10), B(10)
C      NOTE. THE DIMENSION -10- DOES NOT NEED TO AGREE WITH THE CALLING

C     INPUT-
C      X, Y     PTS. ON CURVE
C      NPTS     NO. OF X
C      NEW      =1 (.TRUE.) TO FIT CURVE,  =0 (.FALSE.) TO USE LAST FIT
C      XC       LIST OF X AT WHICH CALC TO BE DONE
C      YC(1)    INTEGRATION CONSTANT IF  ND=-1
C      NXC      NO. OF XC
C      ND       =0 TO GET COORD,  =1 OR 2 TO GET 1ST OR SECOND DERIV.
C               =-1 FOR INTEGRATION
C     OUTPUT
C      YC       COORDINATE OR DERIVATIVE AT XC     OR
C               YC(IC)= INTEGRAL(Y*DX) FROM XC(1) TO XC(IC) WHERE IC=2,NXC
C      B(I)     FIRST DERIVAT AT POINT I (I=1,N)


C     NOTES-
C      -X- MAY BE IN EITHER ASCENDING OR DESCENDING ORDER.
C      FOR INTEGRATION -XC- MUST BE IN THE SAME ORDER AS -X-.  FOR INTERP
C         NO SPECIAL ORDER IS REQUIRED.

       LOGICAL WITHIN

C     FIT THE CUBIC SPLINE
       IF(.NOT.NEW) GO TO 100
       CALL CUBE(X,Y,NPTS, B)

C     INTERPOLATE
  100  I     = 1
       DO 150 IC=1,NXC

C     LOCATE APPROPRIATE INTERVAL
       WITHIN=.FALSE.
       NCOUNT=NPTS
       N     = NCOUNT-1
  101  NCOUNT=NCOUNT-1
       IF(NCOUNT.EQ.0) GO TO 120

       F     = (XC(IC)-X(I)) / (X(I+1)-X(I))
       IF(F.GE.0.) GO TO 110

C       F.LT.0.
       IF(I.EQ.1) GO TO 125
       IF(ND.EQ.(-1)) GO TO 120
       I     = I-1
       GO TO 101

  110  IF(F.LE.1.) GO TO 125
```

```
C       F.GT.1.0
        IF(I.EQ.N) GO TO 125
        IF(ND.EQ.(-1)) GO TO 126
  112 I     = I+1
        GO TO 101

  120 CALL ERROR1

C     PRELIMINARY CALCULATIONS FOR INTERPOLATION OR INTEGRATION
  125 WITHIN=.TRUE.
  126 DX    = X(I+1)-X(I)
        DY    = Y(I+1)-Y(I)
        D       = (B(I)+B(I+1)-2.*DY/DX)/(DX*DX)
        C       = (3.*DY/DX-(2.*B(I)+B(I+1)))/DX
        XD    = XC(IC)-X(I)
        L     = ND+2
        GO TO (130,140,141,142),L

C     ND=-1, INTEGRATE
  130 IF(.NOT.WITHIN) XD=DX
        S1      = (Y(I) + (B(I)/2. + (C/3. + D/4.*XD)*XD)*XD)*XD
        IF(WITHIN) GO TO 135
C     -I- IS BEING INCREMENTED TO FIND APPROPRIATE INTERVAL.  HENCE,
C       CUMULATE THE INTEGRAL OF THE ITH INTERVAL.
        SA    = SA + S1
        GO TO 112
C     APPROPRIATE INTERVAL FOUND.   X(I)-XC(IC)-X(I+1)
  135 IF(IC.EQ.1) SA=YC(IC)-S1
        IF(IC.NE.1) YC(IC)=SA+S1
        GO TO 150

C     ND=0,  INTERPOLATE FOR COORDINATES
  140 YC(IC)= Y(I) + (B(I) + (C + D*XD)*XD)*XD
        GO TO 150

C     ND=1,  FIRST DERIVATIVE
  141 YC(IC)= B(I) + (2.*C + 3.*D*XD)*XD
        GO TO 150

C     ND=2,  SECOND DERIVATIVE
  142 YC(IC)= 2.*C + 6.*D*XD

  150 CONTINUE

        RETURN
        END
```

```
*DECK CUFITR
      SUBROUTINE CUFITR(X,NIC,IMID)
*CUFITR          TEMPORARY ROUTINE FOR                              -CUFITR-
C                DETERMINING INFLUENCE COEFFICIENTS
C                FOR INFIELD BOUNDARY POINTS
C                WHICH TERMINATE -PARTIAL ORTHOGONALS-

       DIMENSION X(4)

      COMMON /CATM  / NX,XDIM,G(5,5)
      DIMENSION       Y(4),B(4)

      X3     = X(IMID)

C     SHIFT X-ELEMENTS ABOVE -IMID- TO THE LEFT
      NMOVE  = NIC-IMID
      CALL MOVE(1,X(IMID+1),X(IMID),NMOVE,1)

      NI = NIC - 1
      DO 60 I=1,NI
      DO 50 II=1,NI
   50 Y(II) = 0.
      Y(I)  = 1.
   60 CALL CUFIT(X,Y,NIC-1, .TRUE., X3,G(IMID,I),1,0,B)

C     SHIFT G(IMID,I) TO THE RIGHT FOR I.GT.IMID
      I      = NI
   70 G(IMID,I+1) = G(IMID,I)
      I      = I-1
      IF(I.GE.IMID) GO TO 70
      G(IMID,IMID)=-1.
      RETURN
      END
```

```
*DECK FFINC
      SUBROUTINE FFINC
CFFINC         INFLUENCE COEFFICIENTS ON FAR FIELD BOUNDARY        -FFINC-
      COMMON /CFB    / L,MA,MB,DFB(30)
      COMMON /CFFINC/ GFF(6)
      COMMON /CFRFIN/ DM(4),ZDN1,ZDN25
      COMMON /CIDEX / M,J,MU,MD,ISTAG
      COMMON /CS1    / S1(300)
      COMMON /CZ     / Z(300)
C
    1 M      = MB
      CALL GETIX
      QDS1   = 2./(S1(MD)-S1(MU))
C     COMPUTE INFLUENCE COEFFICIENTS
      GFF(2)= 0.
      GFF(6)= 0.
      IF( MU.EQ.0 .OR. MD.EQ.0 ) GO TO 20
      GFF(3)= -.865*QDS1
      GFF(4)= -2.*GFF(3)
      GFF(5)= GFF(3)

      GO TO 2
   20 GFF(3)= 0.
      GFF(5)=0.
      IF( MD.EQ.0 ) GO TO 25
      DS1    = S1(MD)-S1(M)
      GFF(5)= -.865/DS1
      ZL     = Z(M)-ZDN1
      RATIO = ((ZL-DS1)/ZL)**2
      GFF(4)= GFF(5)*(RATIO-2.)
      GO TO 2
   25 DS1    = S1(M)-S1(MU)
      GFF(3)= -.865/DS1
      ZL     = ZDN25-Z(M)
      RATIO = ((ZL-DS1)/ZL)**2
      GFF(4)= GFF(3)*(RATIO-2.)
    2 RETURN
      END
```

```
*DECK IAD
      SUBROUTINE IAD
*IAD          IMPLICIT ALTERNATING DIRECTION ROUTINE--STC                -IAD-
C     INPUT
C      MLB,MUB,LO,LNEXT---STATION TABLE
C      B(M) = INDICATOR  (B.GT.0--SUBSONIC)  (B.LE.0 -- SUPERSONIC)
C      A1,A2,A3,A4,A5,A6,A7,A8 = INFLUENCE COEFFICIENTS
C      RHS(M) = RIGHT HAND SIDES OF MATRIX EQUATION
C      A4(M) = 1. FOR FIRST POINT OF DOUBLE STREAMLINE
C     IADM  = -1   LINE RELAXATION ALONG STREAMLINE
C     IADM  =  0   ALTERNATING ORTHOGONAL, STREAMLINE RELAXATION
C     IADM  =  1   LINE RELAXATION ALONG ORTHOGONAL

      COMMON /CB     / B(300)
      COMMON /CDS2   / DS2(300)
      COMMON /CRHS   / RHS(300)
      COMMON /CA2    / A2(300)
      COMMON /CA3    / A3(300)
      COMMON /CA4    / A4(300)
      COMMON /CA5    / A5(300)
      COMMON /CA6    / A6(300)
      COMMON /CA7    / A7(300)
      COMMON /CA8    / A8(300)
       DIMENSION A1(300),A0(300)
       EQUIVALENCE     (A1,A5),(A0,A6)
      COMMON /CDDS2 / DDS2
      COMMON /CIADIN/ RHOBAS,RHOAMP,IADM
      COMMON /CLBL   / LBL,LSS(2),LBLDUM(5)
      LOGICAL          LBL
      COMMON /CMAXIT/ MAXIT,NREFIN,DUMIT(2)
      COMMON /CPI    / PI,DUMPI(5)
      COMMON /CTOLRL/ TOLRL,MAXSWP,CLEN,DS2MX,TOLES2,NSWP,DTOLRL(6),
     *                SGIMIN,TOLINR
C     STATION TABLE
C      INDEX- L=LO,LESTA
C      SCHOKE= STATION CHOKE INDICATOR (ADJWF,BRHS,WRIOUT)
C      MCL    = SHARP CORNER INDICATOR (BLDTBS)
C      MCL    = FIELD INDEX OF CONTROL STREAMLINE (PTMOVE,FLOBAL)
      COMMON /CHDATA/ X1(1),LNEXT(1),MLB(1),MUB(1),PRIM(1),
     1                TYPELB(1),NAMELB(1),ILB(1),FLB(1),S1LB(1),
     1                TYPEUB(1),NAMEUB(1),IUB(1),FUB(1),S1UB(1),
     3                VMB(1),DWDV(1), X2CL(1),VCL(1),MCL(481)
       LOGICAL          PRIM
       INTEGER TYPELB,TYPEUB
       DIMENSION        SCHOKE(1)
       EQUIVALENCE      (SCHOKE,DWDV)
C     TABLE OF INDEX LIMITS
      COMMON /IXORIG/ LHO,LHE, LBDO,LBDE, LTO,LTE, LWO,LWE, LFO,LFE,
     *                LO,LESTA,  LDUM(8),
     *                MO,NM, NJ,NFCOLS, MAXNJ,MAXOL,MAXNM,MAXLE,
     *                LEO,LEE, LRO,LRE,LRD
       DIMENSION        LIMITS(24)
       EQUIVALENCE      (LIMITS,LHO)
      COMMON /CM     / JMS(300)
      COMMON /CIDEX  / MM2,JS,M1,MDN,ISTAG
      COMMON /CIDEXR/ M,MJ1(4),M3,MJ2(4),M5,MJ4(4),M2,MJ5(4),M6,MJ6(4)
      COMMON /ERASE2/ AA4(128),AA8(128),BB(128),A41(128),A42(128),
     *                MSAVE(128),DRASE(732)
```

```
      COMMON /CPRINT/ PDUM(6),PRT(20)
      COMMON /CSS   / SSFML,SSEF,SSEANG,SSDF,SSFEND,SSFND1
     1               ,SSDLE,A4FACT,BRLX,CURRLX,TSIC,RHOC,RHOCSS
       INTEGER        SSFML
       LOGICAL              SSEF,       SSDF,   SSDLE


C     INITIALIZE DS2 TO 0.,  NSWP=0
      CALL SETM(1,0.,DS2,NM)
      NSWP  = 0
      ASSIGN 235 TO LGO
      ALIM  = SQRT( FLOAT(NM) )
      LIMSWP= MAXSWP-IFIX(ALIM)-2
      FNM   = 1./ALIM
      CLENX = 4.*SG1MIN
      ITYPE = IADM+2
      XXK   = 0.
      RHO   = RHOBAS

C     LOOP TO SWEEP THROUGH STATIONS
      LSTART= LO
      LEND  = LESTA
      IF( .NOT.LBL ) GO TO 1
      IF( LSS(2).EQ.0 .OR. LSS(2).LT.LSS(1) ) RETURN
C     SET LIMITS FOR LINE BY LINE SUPERSONIC SOLUTION
      ITYPE = 2
      LSTART= LSS(1)
      LEND  = LSS(2)+1
    1 L     = LSTART
      DS2MX = 0.
      DDS2  = 0.
      IF( RHOAMP.EQ.0. ) GO TO 1111

C     COMPUTE RHO --ITERATION FACTOR
      XXK   = XXK+1.
      IF( XXK.GE.ALIM ) XXK=1.
      TSIN  = SIN(.5*XXK*PI*FNM)
      RHO   = RHOBAS+2.*RHOAMP*TSIN**2
 1111 RHO1  = 1.-RHO
      GO TO (200,2,2) , ITYPE
C     LOOP ACROSS STREAMLINES
    2 MA    = MLB(L)
      MB    = MUB(L)
      IF(NSWP.GE.LIMSWP) PDUM(3)=1.
      M     = MA
    3 K     = 0
    4 K     = K+1

C     BUILD COEFFICIENT TABLES FOR TDSEQ ON ORTHOGONAL
C     GET M2,M3,M5,M6 INDICES
      CALL GETRLX
C     CALCULATE MODIFIED RIGHT HAND SIDES
      IF( B(M).LE.0. ) GO TO 20

C     SUBSONIC BRANCH
   10 AA41  = -(A2(M)+A3(M)+A5(M)+A6(M))
      AA42  = A4(M)-AA41
      BB(K) = RHS(M)-(A2(M)*DS2(M2)+A3(M)*DS2(M3)+RHO1*AA41*DS2(M)
```

```
     *                  +A5(M)*DS2(M5)+A6(M)*DS2(M6))
      AA4K   = AA42+RHO*AA41
      GO TO 30

C     SUPERSONIC BRANCH       ------GET INDEX-- M1
C     SPECIAL 5 POINT CUBIC-- SSFML=3, PICK UP A0
   20 M1     = M2
   21 MM2    = M1
      CALL GETIX
      IF( M1.EQ.0 ) M1=M
      IF( ISTAG.EQ.3 ) GO TO 21
      M1SAV = M1
   25 MM2    = M1
      CALL GETIX
      IF( M1.EQ.0 ) M1=M
      IF( ISTAG.EQ.3 ) GO TO 25
      M0     = M1
      M1     = M1SAV
      AA41   = -(A2(M)+A3(M)+A1(M)+A0(M))
      AA42   = A4(M)-AA41
      BB(K) = RHS(M)-(A1(M)*DS2(M1)+A2(M)*DS2(M2)+A3(M)*DS2(M3)+RHO1*
     *                  AA41*DS2(M) +A0(M)*DS2(M0))
      AA4K   = AA42+RHO*AA41
      IF(SSFML.EQ.3) GO TO 29

C     TRIDIAGONAL DECOMPOSITION
C     IF A6(M)=0, ADJUST LOCALLY TO RHO=1
   30 IF( A6(M).NE.0. ) GO TO 31
   29 BB(K) = BB(K)+RHO1*AA41*DS2(M)
      AA4K   = AA4K+RHO1*AA41
   31 IF( K.GE.2 ) GO TO 50
      AA8(K)= A8(M)/AA4K
      BB(K) = BB(K)/AA4K
      GO TO 61

C     FORWARD DECOMPOSITION
C     SPECIAL LOGIC FOR 2-ND OF DOUBLE POINTS
   50 IF(A4(M).NE.1.) GO TO 51
      GO TO 60
   51 IF(A4(M-1).NE.1.) GO TO 60
      IF( B(M).LE.0. ) GO TO 52
      AA41   = -(A2(M)+A3(M)+A5(M)+A6(M))
      GO TO 53
   52 AA41   = -(A2(M)+A3(M)+A1(M))
   53 AA42   = A4(M)-AA41
      AA4K   = AA42+RHO*AA41
      IF( A6(M).EQ.0. .OR. (B(M).LE.0. .AND. SSFML.EQ.3 ) )
     *AA4K   = AA4K+RHO1*AA41
      AA4K   = 1./(AA4K+A7(M)*AA8(K-2)*AA8(K-1))
      AA8(K)= A8(M)*AA4K
      BB(K) = (BB(K)-A7(M)*(BB(K-2)-AA8(K-2)*BB(K-1)))*AA4K
      GO TO 61
   60 AA4K   = 1./(AA4K-A7(M)*AA8(K-1))
      AA8(K)= A8(M)*AA4K
      BB(K) = (BB(K)-A7(M)*BB(K-1))*AA4K
   61 IF( M.GE.MB ) GO TO 62
      M      = M+1
      GO TO 4
```

```
      DS2(M)= BB(K)

C     BACK SUBSTITUTION
   70 M      = M-1
      K      = K-1
      IF( M.LT.MA ) GO TO 100
      BB(K) = BB(K)-AA8(K)*BB(K+1)

C     CALCULATE DDS2,DS2MX
   62 DDS2   = AMAX1( DDS2,ABS(BB(K)-DS2(M)) )
      DS2(M)= BB(K)
      DS2MX = AMAX1( DS2MX,ABS(DS2(M)) )
      GO TO 70

C     INDEX TO NEXT STATION
  100 IF( DS2MX.GT.CLENX ) CALL ERROR1
      L      = L+LNEXT(L)
      IF( L.LT.LEND ) GO TO 2
C     INCREMENT SWEEP COUNTER
      NSWP   = NSWP+1
      IF( PDUM(3).NE.0. ) CALL TABPRT(5HDS2-A,DS2,NM,NJ)
      IF( PDUM(3).NE.0. ) WRITE (6,999) DDS2,DS2MX
  999 FORMAT(//6X,5HDDS2=,1PE16.8,6X,6HDS2MX=,E16.8//)
      IF( IADM.EQ.1 .OR. LBL ) GO TO 321

C     LOOP TO SWEEP CROSS-STREAM ALONG STREAMLINES
C     NOTE*** ISTAG=3 POINTS ARE SKIPPED
  200 J2     =NJ
      DS2MX = 0.
      DDS2   = 0.
  202 M      = MBEGIN(J2)
C     CONSTRUCT MATRIX COEFFICIENTS ALONG STREAMLINE
      K      = 0
  203 K      = K+1
C     GET INDICES M2,M3,M5,M6
  205 MSAVE(K)= M
      CALL GETRLX
C     IF B(M).LE.0.--(SUPERSONIC-- SUBTRACT A1*DS2(M1) FROM BB
C     IF SSFML.EQ.3 ALSO SUBTRACT A0*DS2(M0) FROM BB
      A41K   = -(A2(M)+A3(M)+A5(M)+A6(M))
      IF( B(M).LE.0. ) A41K=A41K+A5(M)+A6(M)
      A42K   = A4(M)-A41K
      AA4K   = A41K+RHO*A42K
      MDB    = M-1
      IF( A4(M-1).EQ.1.) MDB=M-2
      BB(K) = RHS(M)-(A7(M)*DS2(MDB)+RHO1*A42K*DS2(M)
     *                +A8(M)*DS2(M+1))
      IF( B(M).GT.0. ) GO TO 206
      M1     = M2
 2051 MM2    = M1
      CALL GETIX
      IF( M1.EQ.0 ) M1=M
      IF( ISTAG.EQ.3 ) GO TO 2051
      M1SAV = M1
 2052 MM2    = M1
      CALL GETIX
      IF( M1.EQ.0 ) M1=M
      IF( ISTAG.EQ.3 ) GO TO 2052
```

```
      M0     = M1
      M1     = M1SAV
      BB(K) = BB(K)-A1(M)*DS2(M1)-A0(M)*DS2(M0)


C     PENTA-DIAGONAL MATRIX-- DECOMPOSITION
C     ADJUST TO RHO=1 IF A7(M)=0.
  206 IF( A7(M).NE.0. ) GO TO 207
      BB(K) = BB(K)+RHO1*A42K*DS2(M)
      AA4K  = AA4K+RHO1*A42K
  207 IF( K.GT.2 ) GO TO 220
      GO TO (208,210) , K
  208 CM     = 1./AA4K
      A41(K)= A5(M)*CM
      IF( B(M).LE.0. ) A41(K)=0.
      A42(K)= A6(M)*CM
      IF( B(M).LE.0. .AND. SSFML.EQ.3 ) A42(K)=0.
      BB(K) = BB(K)*CM
      GO TO 225
  210 CM     = 1./(AA4K-A3(M)*A41(K-1))
      A41(K)= (A5(M)-A3(M)*A42(K-1))*CM
      IF( B(M).LE.0. ) A41(K)=A41(K)-A5(M)*CM
      A42(K)= A6(M)*CM
      IF( B(M).LE.0. .AND. SSFML.EQ.3 ) A42(K) = A42(K)-A6(M)*CM
      BB(K) = (BB(K)-A3(M)*BB(K-1))*CM
      GO TO 225
  220 CMA    = A3(M)-A2(M)*A41(K-2)
      CM     = 1./(AA4K-A2(M)*A42(K-2)-CMA*A41(K-1))
      A41(K)= (A5(M)-CMA*A42(K-1))*CM
      IF( B(M).LE.0. ) A41(K)=A41(K)-A5(M)*CM
      A42(K)= A6(M)*CM
      IF( B(M).LE.0. .AND. SSFML.EQ.3 ) A42(K) = A42(K)-A6(M)*CM
      BB(K) = (BB(K)-A2(M)*BB(K-2)-CMA*BB(K-1))*CM
  225 IF( M5.EQ.M ) GO TO 230
      M      = M5
      GO TO 203

C     BACK-SUBSTITUTION LOOP
  230 ASSIGN 231 TO JGO
      GO TO 250
  231 K      = K-1
      ASSIGN 240 TO JGO
      BB(K) = BB(K)-A41(K)*BB(K+1)
      M      = MSAVE(K)
      GO TO 250
  240 K      = K-1
      M      = MSAVE(K)
      IF( K.LT.1 ) GO TO 300
      BB(K) = BB(K)-A41(K)*BB(K+1)-A42(K)*BB(K+2)

C     CALCULATE DDS2,DS2MX
  250 DDS2   = AMAX1(DDS2,ABS(BB(K)-DS2(M)) )
      DS2(M)= BB(K)
      DS2MX = AMAX1(DS2MX,ABS(DS2(M)) )
      GO TO JGO , (231,240)
  300 IF( DS2MX.GT.CLENX ) CALL ERROR1
      J2     = J2-1
      IF( J2.GE.1 ) GO TO 202
```

300

```
      IF( PDUM(3).NE.0. ) CALL TABPRT(5HDS2-B,DS2,NM,NJ)
      IF( PDUM(3).NE.0. ) WRITE (6,999) DDS2,DS2MX

C     INCREMENT SWEEP COUNTER
  320 NSWP  = NSWP+1
C     STREAMLINE SWEEP COMPLETE-- CHECK CONVERGENCE
  321 IF( DDS2.LE.TOLRL*DS2MX ) GO TO 900
      IF( NSWP.LE.MAXSWP ) GO TO 1
      ASSIGN 234 TO LGO
  900 IF( PRT(2).EQ.0.) GO TO 902
      CALL TABPRT(5HDS2-F,DS2,NM,NJ)
  902 GO TO LGO , (234,235)
  234 CALL ERROR1
  235 RETURN
      END
```

```
*DECK SS5PTI
      SUBROUTINE SS5PTI(XX,G)
*SS5PTI          SUPERSONIC 5-PT INFLUENCE COEFFICIENTS       -SS5PTI-
      DIMENSION        XX(5),G(25)

C     INPUT-
C       XX     = STREAMWISE DISTANCE OF FOUR POINTS, XX(1),..X(4)

C     OUTPUT-
C       G      = CHANGE IN SECOND DERIVATIVE, D2YDX2, PER UNIT CHANGE IN
C                YY(0),...Y(4)

      COMMON /CSS5PT/ X(4),Y(4), X21,X31,X32,X41,X42,X43, A0,A1,A2,A3,A4

C     X(0)   = 0.
      DO 65 I=1,4
   65 X(I)   = XX(I+1)-XX(1)
      CALL SS5PT
      G(5)   = A0
      G(10)  = A1
      G(15)  = A2
      G(20)  = A3
      G(25)  = A4
      RETURN
      END
```